Contract #W7714-4-9858

(document MD002-06)

TAPE INTERFACE BOARD
MAINTENANCE MANUAL

written by:

Pierre Da Sylva

MultiDev JDL Communications Inc.
1040 St.Joseph Blvd., suite 8
Hull, Quebec J8Z 1X9

January 18, 1996

REVISION SHEET

REVISION	DATE	AUTHOR
MD002-06	January 18, 1996	Pierre Da Sylva
Approval: Jean Caseault		

GLOSSARY

C51	*Concept 51 Disk Array. Refer to [3]*
CPU:	*Central Processing Unit.*
DFB:	*Differential Fast Bus*, used by the disk array.
DLCS:	*Data Logger and Control System.* The VME-based data acquisition system.
DOS:	*Disk Operating System,* from Microsoft.
EARS:	*Experimental Array Radar System.*
FIFO:	*First-in, First-out* memory.
IF:	*Intermediate Frequency*
I/O:	*Input/Output*
LED	*Light Emitting Diode*
OS-9	*Operating System*
PC:	*Personal Computer.*
SCSI:	*Small Computer System Interface*
TIB:	*Tape Interface Board.*
VLDS:	*Very Large Data Storage.*
VMEHD	*VME to C51 Interface Card. Refer to [3]*

TABLE OF CONTENTS

1. INTRODUCTION 6
2. SYSTEM REQUIREMENTS 7
3. BOARD INTERCONNECT 8
3.1 DISK CONNECTION 8
3.2 TAPE CONNECTIONS 9
3.2.1 Control 9
3.2.2 Data 9
4. BOARD PREPARATION 10
5. DIAGNOSTICS 11
5.1 POWER UP 11
5.2 FIFO MEMORY 12
5.3 DISK INTERFACE 13
5.3.1 Writing to Disk 13
5.3.2 Verifying Disk Data 13
5.3.3 Example 14
5.4 TAPE INTERFACE 15
5.4.1 Tape Unit Reset 15
5.4.2 Tape Unit Command 15
5.4.3 Tape Recording 16
5.4.4 Tape Reproduce 16
5.4.5 Example 17
5.5 DISK BACK-UP 18
5.5.1 Disk to Tape 18
5.5.2 Tape to Disk 19
5.5.3 Example 19
6. REFERENCES 22
APPENDIX A: PARTS LIST 23
APPENDIX B: SCHEMATICS AND DRAWINGS 25
APPENDIX C: INTERCONNECT 26
APPENDIX D: BOARD MODIFICATIONS 31

LIST OF FIGURES

Figure 1: TIB Inter-connect 8
Figure 2: Address Switch Settings 10
Figure 3: Front Panel LED's 11

1. INTRODUCTION

This manual contains the information required to ensure the proper installation and operation of the Tape Interface Board. The aim of this manual is to take the operator through a series of steps that, if completed successfully, will indicate a properly functioning board. If at any point in the procedure, improper operation is detected, or sample listings differ from the ones given by the card under test, refer to the manufacturer for servicing.

2. SYSTEM REQUIREMENTS

For these tests to have significance, the Tape Interface Board must be installed in a VMEbus chassis with a system controller running OS-9 (Ver 2.4). The required software is contained in the floppies accompanying the manual. There must also be a Metrum VLDS tape unit [4]and a Storage Concept C51 disk unit [3](with accompanying VMEHD disk interface card) attached to the Tape Interface Board. It is recommended that the EARS data acquisition system be used.

NOTE: For these tests, it is recommended that a specific hard disk array system be used. This is due to changing specs from the manufacturer (Storage Concepts) on more recent devices. The system to use should have the Model #:051R-09021NM-00; Serial number: 1237 [3].

3. BOARD INTERCONNECT

The card should be connected in the system as per Figure 1.

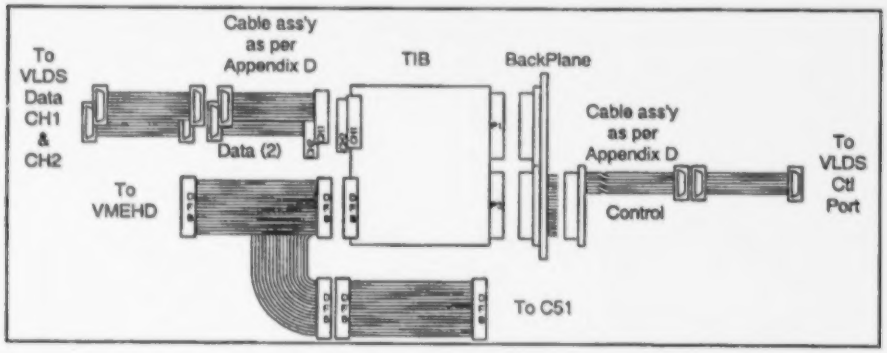


Figure 1: TIB Inter-connect

When the card is installed in the EARS system, the front panel and P2 cable extensions should already be available. If not, the P2 cable extension should be installed on the backplane. Due to changing configurations, care must be taken as to which slot the cable extension should be installed. It is recommended that the card be installed in the first right-most open slot in the backplane. The P2 cable extension is then placed, on the backplane, at the same slot level as the slot number of the card, on P2.

The Differential Fast Bus cabling should be daisy-chained on the front panels of the VMEHD and the Tape Interface Board, then connecting through to the back panel of the system. To complete the connections, connect the data and control cables to the Metrum Tape Unit.

3.1 DISK CONNECTION

The Tape Interface Board is connected to the DFB between the VMEHD (in the system's card cage) and the system's back panel. The back panel connectors to the C51 (HIB and DFB) should already be connected to the disk unit when using the EARS system. Because the Tape Interface Board is inserted between the two ends of the DFB chain [3], termination for this bus is not required. The Tape Interface Board was designed to operate as an end card, so termination resistors were designed in. However, for this application, they were not installed (see Appendix D for details).

3.2 TAPE CONNECTIONS

The Tape Interface Board is connected to the Metrum tape unit through two ports: the control and data ports. The control port is used to send commands to the tape unit for normal operation. Data is stored on tape via the data port. The control port uses a DB25F connector. The data port uses two DB37F connectors.

3.2.1 Control

The tape control port is connected through P2 of the Tape Interface Board to the system's back panel, using the cable assembly described in Appendix D. The connection to the tape unit is made using a 25 conductor DB25F to DB25M ribbon assembly. Use J4 on the tape unit.

3.2.2 Data

The data ports are connected using two 40-pin ribbon to DB37M cable assemblies (as per Appendix D), going to the system's back panel. Connections to the tape unit are made using DB37F to DB37M ribbon cable assemblies. Ch1 data connects to J3 of the tape connectors, Ch2 data connects to J2 of the tape unit.

4. BOARD PREPARATION

For the card to work properly, the address of the Tape Interface Board on the VMEbus must be set. This is done by setting SW1 to the following:

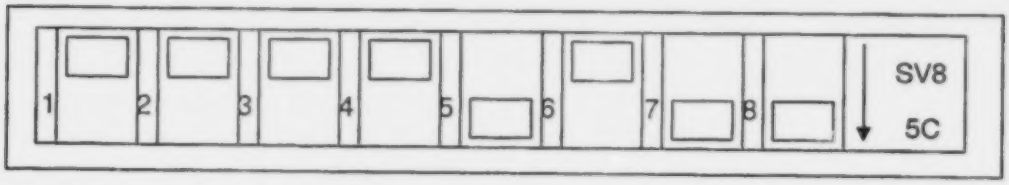


Figure 2: Address Switch Settings

This operation sets the base address of the card at: FE200000h.

5. DIAGNOSTICS

The following section describes the different functions developed for the debugging of the Tape Interface Board. In order to accomplish these, a terminal must be installed on the console port of the system controller (the Heurikon card). The port settings are: **9600 Baud, 8 bits, no parity**. Connections to the terminal are straight through.

To execute any of the described commands, simply login to OS-9 and at the OS-9 prompt enter the commands as they appear in the document.

For the following operations, any transfer occurring to tape or to disk, the transfer LED's (VLDS, DFB) on the front panel should turn green. In idle mode, these indicators are red (Figure 3).

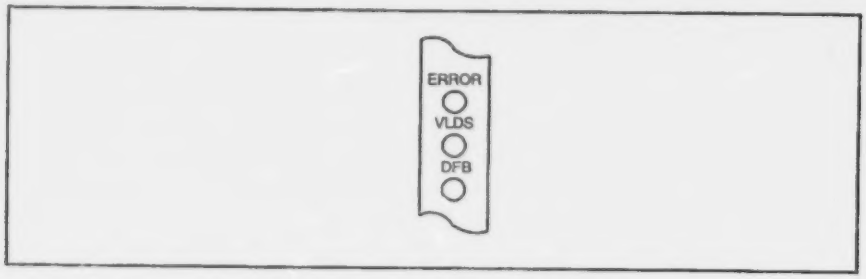


Figure 3: Front Panel LED's

5.1 POWER UP

Power up is indicated on the front panel by the lighting of the three LED's in green. If proper reset is accomplished, these should then go back to the red color. The Error LED is red due to the signaling of the VLDS (using the ATTN0- line), following a power up. If the VLDS is not in operation (powered down), the Error LED should go to red, indicating that the tape unit is not selected and the ATTN0- line is floating (Low). To return the Error LED to green, turn on the tape unit and send the SELECT command to the tape, followed by a READ STATUS command (section 5.4.2).

5.2 FIFO MEMORY

The FIFO buffer of the Tape Interface Board can be tested by using the following command:

tibmem

This software will load a ramp into the FIFO buffer, then read it back while comparing it to the original pattern. During that time, the buffer flags are tested to verify that they are in the proper state. The VLDS LED may turn green briefly while the test is performed.

The following listing gives an example of the messages that could be displayed:

```
OS9>> tibmem
The TIB has been reset
The FIFOs are empty, and they are not full
Written 1 word to the FIFOs
The FIFOs are no more empty, and they are not full
Almost filled the FIFOs ...
The FIFOs are not empty, and they are not full
Written last word to the FIFOs
The FIFOs are not empty, and they are full
First reading matches the expected pattern
The FIFOs are not empty, and they are no more full
Almost emptied the FIFOs ...
The FIFOs are not empty, and they are not full
Last reading matches the expected pattern
The FIFOs are now empty, and they are not full
OS9>>
```

During the execution of this task, the FIFO flags are being monitored. If, at any moment, the status of the flags is not as it should be, the listing will differ from the one given here: messages should appear in uppercase. If this occurs, please contact the manufacturer.

5.3 DISK INTERFACE

The disk interface is verified by a two-step procedure. First, a block of data is written to the disk using the VMEHD interface card. To verify the disk interface on the Tape Interface Board, a read command performs the transfer of the same data block from the disk to the system controller via the Tape Interface Board. The pattern written by the first command is verified in the second command.

Accesses made to the disk via the DFB cause the DFB LED to turn green momentarily.

5.3.1 Writing to Disk

To verify that the disk interface is working properly, a pattern is first written to the disk using the following command:

wrc51 <word_start> <word_increment> <first_sector> <num_frames>

where: **word_start** and **word_increment** are 16-bit words, in decimal
first_sector is the first 18 KB sector of the file
num_frames is the number of 288-Kword frames to write

The pattern consists of a ramp starting at **word_start**, counting up by **word_increment**. The data is stored starting at **first_sector** for **num_frames** of 288 Kwords (2 x 9 disks x 16 Kwords/disk). Specifying the number of frames is useful when a large number of data frames needs to be verified or when a disk to tape transfer needs testing. For initial debugging, a smaller number is used.

5.3.2 Verifying Disk Data

Once data has been written to disk using the command in the previous section (5.3.1), this command retrieves the data from the disk unit using the Tape Interface Board:

repc51 <word_start> <word_increment> <first_sector> <num_frames>

where: **word_start** and **word_increment** are 16-bit words, in decimal
first_sector is the first 18 KB sector of the file
num_frames is the number of 288-Kword frames to read

The pattern generated must be the same as specified when writing to the disk (section 5.3.1). If no errors are detected, the disk interface is operating properly.

5.3.3 Example

The next section describes the steps to follow to verify the disk interface.

The first steps involves writing the pattern to disk:

wrc51 1 2 2 1

The pattern starts with **1** and is incremented by **2**. The sequence is stored starting at sector **2**, for **1** 288 Kword block.

Next, the data is read back using the Tape Interface Board. The Tape Interface Board buffer is read by the software and compared to the same pattern that was stored in the first command. This test is performed when the following command is issued:

repc51 1 2 2 1

If everything is operating as planned, the following listing should be generated:

```
OS9>> wrc51 1 2 2 1
Configuring VME
Configuring C51 initial state
Configuring C51 operating mode
Writing to C51
1 frames written to disk
OS9>> repc51 1 2 2 1
Configuring VME
Configuring C51 initial state
C51 is now configured
The FIFOs are empty, and they are not full
C51 has been told how much data to read
January 10, 1996 Wednesday 4:56:24 pm
The FIFOs are not empty, and they are full
The FIFOs are now empty, and they are not full
January 10, 1996 Wednesday 4:56:27 pm
OS9>>
```

5.4 TAPE INTERFACE

The following section describes the steps to follow in order to determine the status of the Tape Interface Board tape control circuitry. By performing these steps, the following functions are tested: the tape reset control, the tape command controls, the recording on tape and the verification of the recording by reading the stored data.

For this test, it is highly recommended that a new blank tape be used. A single tape can be used for all tests.

Accesses made to the tape through the data port will cause the VLDS LED to turn green momentarily.

5.4.1 Tape Unit Reset

Special circuitry was designed to perform a reset of the Metrum Tape Unit. A reset pulse of 1 μsec must be applied to the RESET0- line of the command port to accomplish this. The tape is reset by using the following command:

wvlds 99

Following this command the tape goes through a procedure where the LEDs on the front panel of the tape unit are flashing momentarily, all off first, then all on, then all off again. Afterwards, the power light stays on, followed by the selected/access light. This last one turns on because the SELECT command (5A) is issued with this operation. If the selected light does not turn on and the reset has occurred anyway, the command interface is not working properly. If a tape is loaded in the tape unit, the LOADED light turns on.

Because the tape has been reset, the attention signal (ATTN0-) is being asserted. This is indicated on the Tape Interface Board by the Error LED being red. To return the Error LED to its proper color (green), simply read the tape status register '0' (as per section 5.4.2).

5.4.2 Tape Unit Command

To send a command to the tape unit, the following syntax is used:

wvlds <command_number> [<command_parameters>]

where: **command_number** is a two-digit hex number
command_parameters is a string of 1 to 12 ASCII numeric characters representing a decimal number

command_number and **command_parameters** refer to the Metrum tape command syntax (refer to the Metrum Technical Manual for more details) [4].

For example, to reproduce 5 file marks, use:

wvlds 6C 5

This syntax must be used for all commands issued to the tape except for READ STATUS (51). To read the status registers of the tape unit, the following command must be used:

rvlds <status_register_number>

where: **status_register_number** is a decimal number

status_register_number refers to a register of the tape unit as specified in section 3-17 of the Metrum Technical Manual.

5.4.3 Tape Recording

To verify that the interface can properly save data to tape, the following command is used:

recvlds <word_start> <word_increment>

where: **word_start** and **word_increment** are 16-bit words, in decimal

The command will first initialize the FIFO buffer (128K x 16) to the specified pattern. The pattern consists of a ramp starting at **word_start**, counting up by **word_increment**. When the pattern is loaded, the Tape Interface Board starts transferring data to the tape after issuing a command to record to the tape. This operation is required to validate the next operation. If this operation is not performed, the data on the current file mark is unknown and cannot be verified easily. When the tape receives the RECORD command (46), it automatically finds the end of data on the tape.

5.4.4 Tape Reproduce

This operation is performed following the recording of a known pattern to tape (see previous section). This command issues a REPRODUCE command to the tape unit, waits for the Tape Interface Board buffers to fill and verifies that the data is as per the specified pattern:

repvlds <word_start> <word_increment>

where: **word_start** and **word_increment** are 16-bit words, in decimal

The pattern consists of a ramp starting at **word_start**, counting up by **word_increment**. If this command performs without any error, the tape interface is considered to be working adequately.

5.4.5 Example

This section describes a sample session for debugging the tape interface. First, a tape reset is issued. Next, the end of data on tape is reached by using the end of tape command. A tape status register read command is also included.

Following this, the Tape Interface Board FIFO buffer is filled with a pattern and then sent to the tape. After returning the tape to the beginning of this new file, the data is retrieved from the tape and verified. The terminal indicates the following messages when performing these steps:

```
OS9>> wvlds 99                          (tape reset)
VLDS has been reset
OS9>>  wvlds 49                         (find end of tape)
Command has been sent to VLDS
OS9>> rvlds 0                           (read status register)
VLDS register #0 contains 1C
OS9>> recvlds 2 3                       (record the pattern to tape)
TIB has been reset
The FIFOs are empty, and they are not full
TIB has been set for data transfer
Filling the FIFOs ...
The FIFOs are full
TIB has been told to start transferring data
VLDS has been told to record data
VLDS has been told to stop recording data
TIB has been told to stop transferring data
128 Kwords have been transferred to the tape successfully
OS9>> wvlds 4c                          (find the beginning of the file)
Command has been sent to VLDS
OS9>> repvlds 2 3                       (read the tape data and verify)
TIB has been reset
The FIFOs are empty, and they are not full
TIB has been told to start transferring data
VLDS has been told to reproduce data
VLDS has been told to stop reproducing data
TIB has been told to stop transferring data
128 Kwords have been read and verified successfully
OS9>>
```

5.5 DISK BACK-UP

The Disk back-up procedure verifies the operation of transferring data from the disk array to the VLDS tape unit. This operation requires four steps:

1) Load the disk with a known pattern
2) Transfer this pattern to the tape
3) Transfer the pattern back to the disk
4) Verify that the transferred pattern is the same as the original.

This first method requires that both disk to tape and tape to disk transfers occur properly. If either operation is not done correctly, performing these operations will not determine which data direction is faulty. Successful disk to tape transfers can be verified in a slightly different fashion. To isolate the directions, a second sequence is used:

1) Load the disk with a known pattern for a data block of 128K x 16
2) Transfer this pattern to the tape
3) Position the tape at the new file mark
4) Verify that the transferred pattern is the same as the original through the tape port.

For these test, both the VLDS and DFB LED's will turn green momentarily during transfers.

5.5.1 Disk to Tape

The first step in verifying the operation of the back-up consists of a series of data block transfers from the C51 disk unit to the Metrum tape unit using the Tape Interface Board. Before this command is used, a pattern must be written on the disk unit to facilitate the verification of the transfers. See section 5.3.1 for details on how to load a known pattern to the disk.

Once the pattern has been written to disk, the following command is used to initiate and perform the transfer to tape (this transfers a file from the C51 to the VLDS through the Differential Fast Bus:

c51vlds <first_sector> <num_frames>

where: **first_sector** is the first 18 KB sector of the file
num_frames is the number of 288-Kword frames in the file

Having completed this step, a second step described below is required to verify that the transfer occurred without a single failure.

5.5.2 Tape to Disk

Tape to disk transfers are required to verify that the transfer from disk to tape has occurred properly. The transfer starts by placing the tape unit at the beginning of the current file. The disk unit is set up for data and the REPRODUCE command is issued to the tape unit. To generate this procedure, the following command is used (this transfers a file from C51 to VLDS through the Differential Fast Bus):

vldsc51 <first_sector> <num_frames>

where: **first_sector** is the first 18 KB sector of the file
num_frames is the number of 288-Kword frames in the file

Once the transfer is complete, the data must be verified on the disk using the **repc51** command (section 5.3.2).

5.5.3 Example

The next example shows a session for backing 1 frame of 2 x 288 KB.

Because the disk frame is 2 x 288 KB long, and the Tape Interface Board buffer is only 128K x 16 deep, only the first 128 KB of data will be read and compared. If the transfer is successful, the screen should look like this:

OS9>> wrc51 1 1 2 1 (write a ramp starting at 1 and incrementing by 1 for 1 frame)
Configuring VME
Configuring C51 initial state
Configuring C51 operating mode
Writing to C51
1 frames written to disk
OS9>> wvlds 49 (find the end of data on the tape)
Command has been sent to VLDS
OS9>> c51vlds 2 1 (transfer the frames to the Metrum)
Configuring VME
Configuring C51 initial state
C51 is now configured
A cassette has been detected
The cassette is writable
C51 has been told to read
TIB has been set for data transfer
VLDS has been told to record
VLDS has been told to stop recording
January 11, 1996 Thursday 4:56:31 pm

OS9>> wvlds 4c (go back to beginning of file, the file mark)
Command has been sent to VLDS
OS9>> repvlds 1 1 (read and verify the frame on tape)
TIB has been reset
The FIFOs are empty, and they are not full
TIB has been told to start transferring data
VLDS has been told to reproduce data
VLDS has been told to stop reproducing data
TIB has been told to stop transferring data
128 Kwords have been read and verified successfully
OS9>>

To verify the disk to tape transfer of 64 frames, the following sequence is used:

This sequence uses the transfer from disk to tape and back to disk sequence for verifying the operation of the Tape Interface Board. The listing on the terminal should look like this:

OS9>> wrc51 1 1 1 64 (write a ramp starting at 1 and increment by 1 for 64 frames)
Configuring VME
Configuring C51 initial state
Configuring C51 operating mode
Writing to C51
64 frames written to disk
OS9>> c51vlds 1 64 (transfer the frames to the Metrum)
Configuring VME
Configuring C51 initial state
C51 is now configured
A cassette has been detected
The cassette is writable
C51 has been told to read
TIB has been set for data transfer
VLDS has been told to record
VLDS has been told to stop recording
January 11, 1996 Thursday 5:14:09 pm
OS9>> wvlds 4c (go back to file mark)
Command has been sent to VLDS

OS9>> vldsc51 1 64 (copy the saved frames back to the disk)

Configuring VME

Configuring C51 initial state

C51 is now configured

A cassette has been detected

C51 has been told to write

TIB has been set for data transfer

January 11, 1996 Thursday 5:15:22 pm

C51 has been told to stop writing

TIB transfer mode has been disabled

VLDS has been told to stop reproducing

January 11, 1996 Thursday 5:15:37 pm

OS9>> repc51 1 1 1 64 (verify 64 frames of data with the specified pattern)

Configuring VME

Configuring C51 initial state

C51 is now configured

The FIFOs are empty, and they are not full

C51 has been told how much data to read

January 11, 1996 Thursday 5:26:13 pm

The FIFOs are not empty, and they are full

The FIFOs are now empty, and they are not full

January 11, 1996 Thursday 5:27:57 pm

OS9>>

6. REFERENCES

[1] *Development of an Enhanced Signal Processing Data Acquisition and Display System*, MultiDev JDL Communications Inc., June 1995.
[2] *Tape Interface Board - Detailed Design Document*, MultiDev JDL Communications Inc., January 1996.
[3] *Differential Fast Bus Interface Manual*, revision 2.0, Storage Concepts, May 1989.
[4] *Metrum Technical Manual*, Metrum, February 1994.

APPENDIX A: PARTS LIST

TIB: FIFO BUFFER Revised: Thursday, September 07, 1995

MX002-01 Revision: A.0

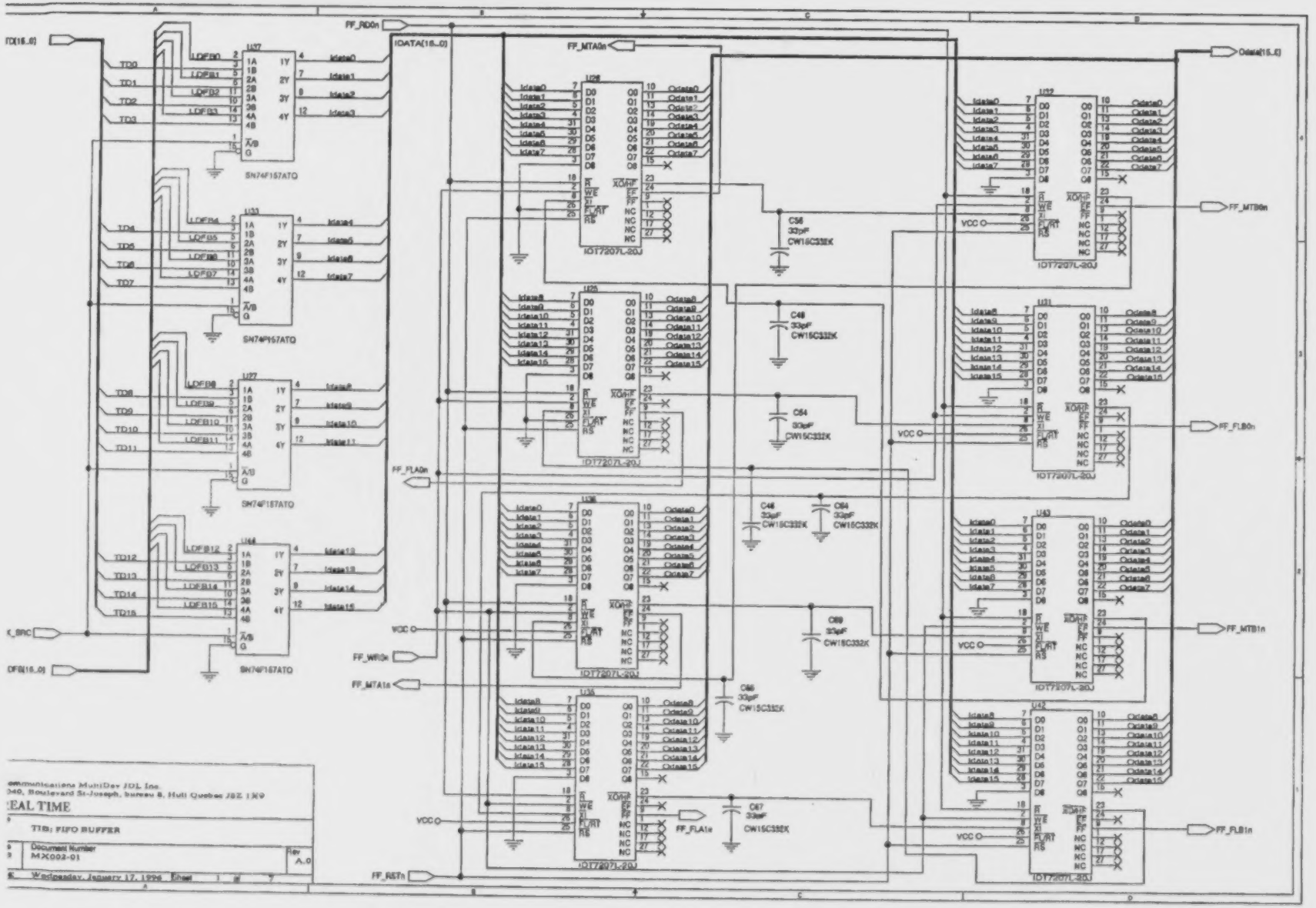
Bill Of Materials January 15, 1996 12:21:19

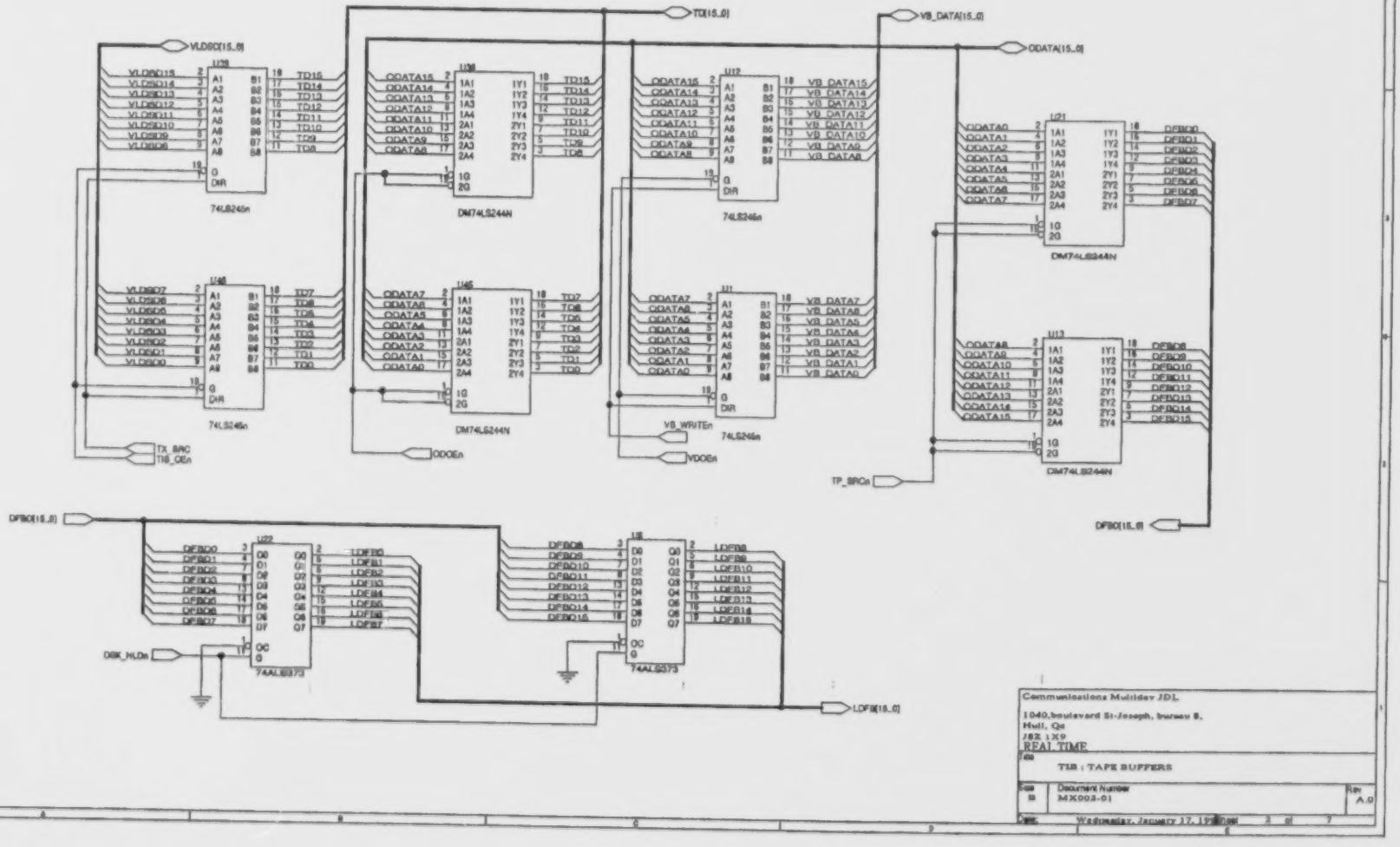
Item	Qty	Reference	Part	Part Number	Manufacturer
1	55	C1, C2, C3, C4, C5, C6, C7, C8, C10, C11, C12, C13, C14, C15, C16, C17, C18, C19, C20, C22, C23, C24, C25, C26, C27, C28, C29, C31, C34, C35, C36, C37, C38, C39, C41, C42, C43, C44, C45, C47, C49, C50, C51, C52, C53, C55, C57, C58, C59, C60, C61, C62, C63, C66, C68	0.01uF	C315C103K1R5CA	Kemet
2	4	C9,C21,C32,C33	1uF	SR305C105KA	Kemet
3	2	C40,C30	10uF	T350K106K050AS	Kemet
4	8	C46,C48,C54,C56,C64,C65, C67,C69	33pF	CW15C332K	Dale
5	1	JP1	HEADER 2x1	4-103322-0 (2p)	Amp
6	1	JP2	HEADER 3X2	4-103322-0 (6p)	Amp
7	1	JP3	DFB_CONN	3433-5302	3M
8	1	JP4	HEADER 4X2	4-103322-0 (8p)	Amp
9	3	LED1,LED2,LED3	2 COLOR LED	550-3509	Dialight
10	1	P1	96 Pin Din	09-03-196-6921	Amp
11	1	P2	96 Pin Din	09-03-196-6921	Amp
12	1	P3	JTAG_CONN	3591-6002	3M
13	1	P4	TAPE_CTL2	3432-L202	3M
14	2	R1,R9	4308R-101-151	4308R-101-151	Bourns
15	1	R2	4116R-001-151	4116R-001-151	Bourns
16	1	R3	4308R-101-472	4308R-101-472	Bourns
17	6	R4,R5,R6,R7,R11,R12	4308R-101-221	4308R-101-221	Bourns
18	4	R8,R10,R16,R17	221K	RN-55D 221K	Dale
19	6	R13,R14,R15,R18,R19,R20	33R	RN-55D 33R0	Dale
20	1	S1	8 SW. SIP	SSV08	Alcoswitch
21	3	TP1,TP2,TP3	test point pin	180-7337-02-05	Cambion
22	4	U1,U12,U39,U46	8 Bit Bi-Di. Buf.	DM74LS245N	National
23	11	U2,U3,U7,U8,U9,U14,U15, U16,U17,U18,U19	MC34050	MC34050P	Motorola
24	2	U4,U11	MACH445	MACH445-12YC	AMD
25	2	U20,U5	One Shot	SN74LS123N	Motorola
26	2	U6,U22	8 Bit Latch	SN74LS373N	Motorola
27	1	U10	OSCILLATOR	F1100-20MHz	Fox
28	5	U13,U21,U24,U38,U45	8 Bit Buffer	DM74LS244N	National
29	2	U23,U30,U34	Comparator	SN74LS682N	Motorola
30	8	U25,U26,U31,U32,U35,U36, U42,U43	32Kx9 FIFO	IDT7207L-20J	IDT

Item	Qty	Reference	Part	Part Number	Manufacturer
31	4	U27,U33,U37,U44	2:1 MUX	SN74F157ATQ	
32	1	U28	MC3487P	MC3487P	Motorola
33	1	U29	VME2000	VME2000P-45	PLX
34	2	U41,U40	8 Bit Latch	QS74FCT543ATP	Quality Semi.
35	1	P.C.B. blank	P.C.B.	MX002-01 REV A.0	Prism Circuits
36	1	P.C.B. bracket	TIB-VME Bracket	BKT002-01 REV A.0	Scanbee

APPENDIX B: SCHEMATICS AND DRAWINGS

1) FIFO Buffer
2) Tape Buffers
3) Tape Interface
4) DFB Interface
5) VMEbus Interface
6) DFB, Buffer Controls and LED Controls
7) Decoupling
8) PCB Silkscreen
9) Front Plate


FF_RDOn
IDATA(15..0)
FF_MTA0n
FF_FLA0n
FF_WR0n
FF_MTA1n
FF_FLA1n
FF_RSTn
VCC
SN74F157ATQ
IDT7207L-20J
C56 33pF CW15C332K
C48 33pF CW15C332K
C54 33pF CW15C332K
C46 33pF CW15C332K
C64 33pF CW15C332K
C69 33pF CW15C332K
C66 33pF CW15C332K
C67 33pF CW15C332K
FF_MTB0n
FF_FLB0n
FF_MTB1n
FF_FLB1n
TTB: FIFO BUFFER
Document Number MX002-01
Rev A.0
Wednesday, January 17, 1996

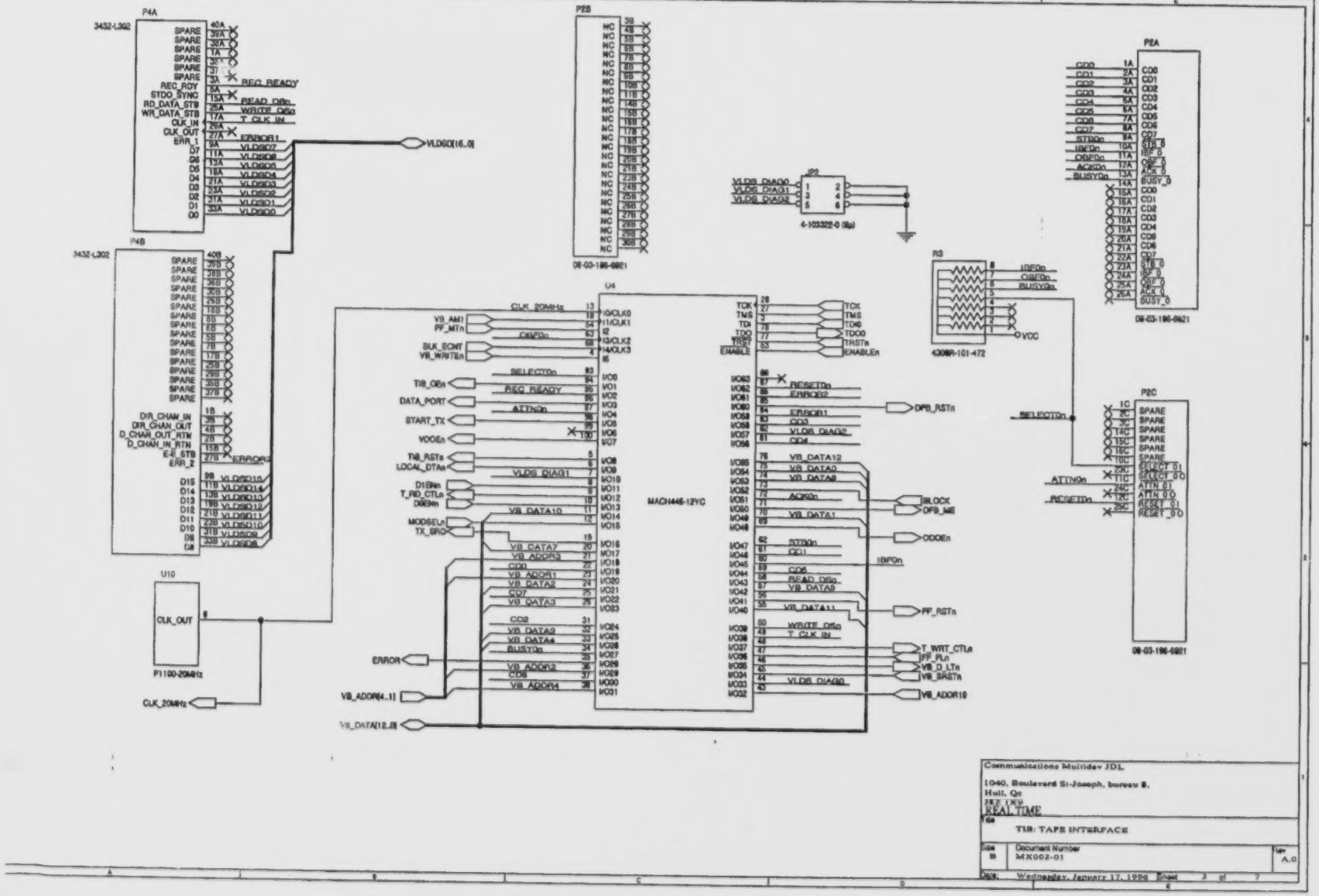


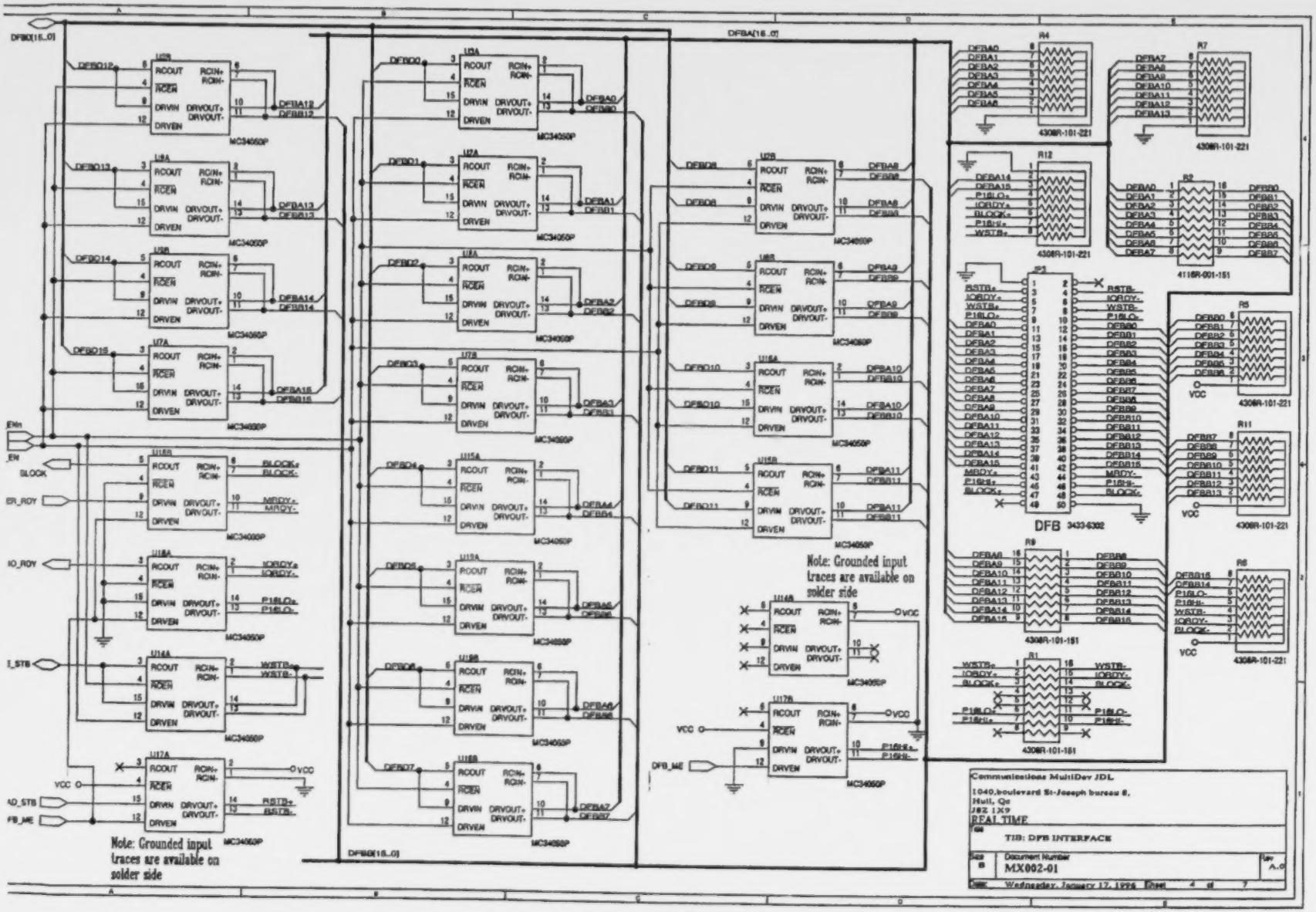
Communications Multidev JDL
1040,boulevard St-Joseph, bureau 8,
Hull, Qe
J8Z 1X9
REAL TIME

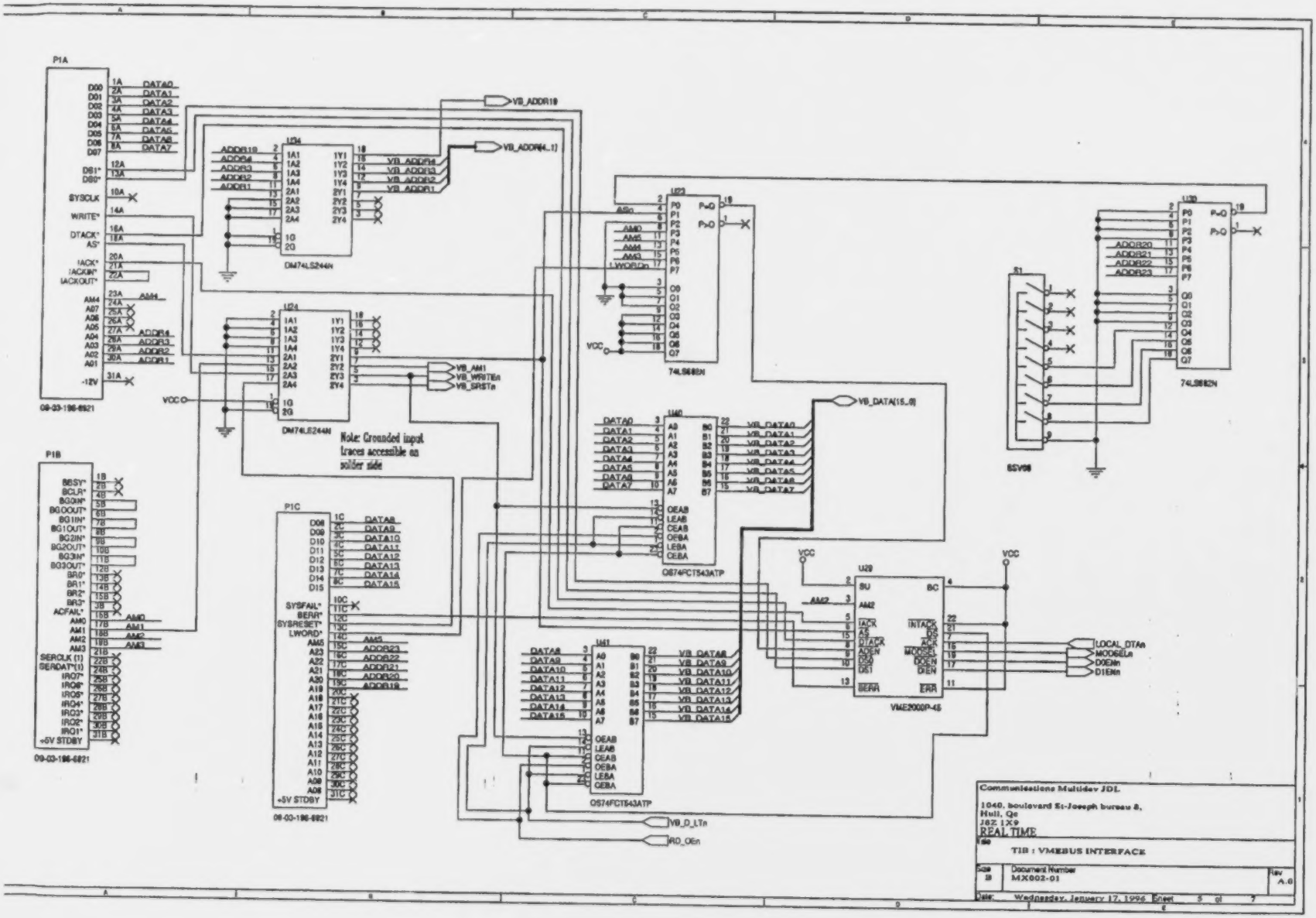
TIB : TAPE BUFFERS

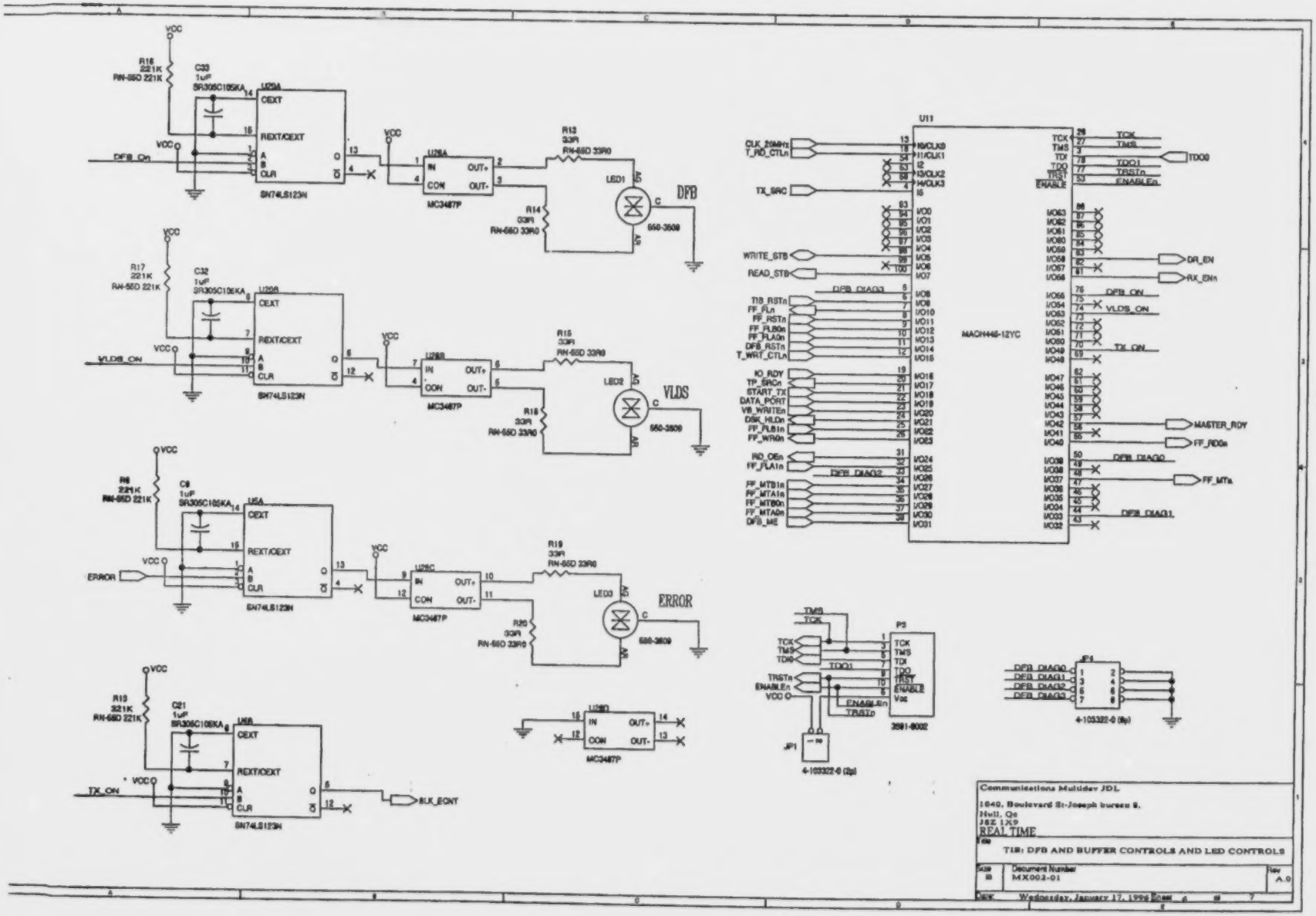
Size: B | Document Number: MX003-01 | Rev: A.0

Date: Wednesday, January 17, 1990 | Sheet 3 of 7


P4A
3432-L302
SPARE
REC_RDY
STDO_SYNC
RD_DATA_STB
WR_DATA_STB
CLK_IN
CLK_OUT
ERR_1
D7
D6
D5
D4
D3
D2
D1
D0
REC_READY
READ_DSo
WRITE_DSo
T_CLK_IN
ERRDS1
VLDSD7
VLDSD6
VLDSD5
VLDSD4
VLDSD3
VLDSD2
VLDSD1
VLDSD0
VLDS[16..0]
P4B
3432-L302
DIR_CHAN_IN
DIR_CHAN_OUT
D_CHAN_OUT_RTN
D_CHAN_IN_RTN
E-E_STB
ERR_2
ERRDS2
D15
D14
D13
D12
D11
D10
D9
D8
VLDSD15
VLDSD14
VLDSD13
VLDSD12
VLDSD11
VLDSD10
VLDSD9
VLDSD8
U10
CLK_OUT
P1190-20MHz
CLK_20MHz
P2B
08-03-186-6821
VLDS_DIAG0
VLDS_DIAG1
VLDS_DIAG2
J22
4-103322-0 (8pl)
U4
MACH445-12YC
CLK_20MHz
VB_AM1
PF_MTn
CENDn
BLK_EOMT
VB_WRITEn
SELECTDn
TIB_OEn
REG_READY
DATA_PORT
ATTNDn
START_TX
VDOEn
TIB_RSTn
LOCAL_DTAn
VLDS_DIAG1
DIBWn
T_RD_CTLn
DOEWn
MODSELn
TX_SRQ
VB_DATA10
VB_DATA7
VB_ADDR3
CD0
VB_ADDR1
VB_DATA2
CD7
VB_DATA3
CD2
VB_DATA9
VB_DATA4
BUSYDn
VB_ADDR2
CD6
VB_ADDR4
ERRDR
VB_ADDR[4..1]
VB_DATA[12..0]
TCK
TMS
TDI
TDO
TRSTn
ENABLEn
RESETDn
ERRDR2
ERRDR1
CD3
VLDS_DIAG2
CD4
VB_DATA12
VB_DATA0
VB_DATA8
ACKDn
VB_DATA1
STRDn
CD1
IBFDn
CD5
READ_DSo
VB_DATA5
VB_DATA11
WRITE_DSo
T_CLK_IN
VB_ADDR10
DPB_RSTn
BLOCK
DPB_ME
ODOEn
PF_RSTn
T_WRT_CTLn
PF_PLn
VB_D_LTn
VB_BRSTn
R3
43089-101-472
IBFDn
OBFDn
BUSYDn
VCC
P2A
08-03-186-6821
CD0
CD1
CD2
CD3
CD4
CD5
CD6
CD7
STRDn
IBFDn
OBFDn
ACKDn
BUSYDn
STB_0
IBF_0
OBF_0
ACK_0
BUSY_0
P2C
08-03-186-6821
SELECTDn
ATTNDn
RESETDn
SPARE
SELECT_01
SELECT_00
ATTN_01
ATTN_00
RESET_01
RESET_00
Communications Multidev JDL
1040, Boulevard St-Joseph, bureau 8.
Hull, Qc
REAL TIME
TIB: TAPE INTERFACE
Document Number
MX002-01
Rev
A.0
Wednesday, January 17, 1996
Sheet 3 of 7


DFB0[15..0]
DFBA[15..0]
MC34050P
4308R-101-221
4116R-001-151
DFB 3433-6302
Note: Grounded input traces are available on solder side
Note: Grounded input traces are available on solder side
Communications MultiDev JDL
1040,boulevard St-Joseph bureau 8,
Hull, Qc
J8Z 1X9
REAL TIME
TIB: DFB INTERFACE
Document Number
MX002-01
Wednesday, January 17, 1996


P1A
P1B
P1C
U34
U24
U23
U30
U40
U41
U29
S1
DM74LS244N
74LS682N
OS74FCT543ATP
VME2000P-4S
09-03-196-6921
VB_ADDR19
VB_ADDR[4..1]
VB_AM1
VB_WRITEn
VB_SRSTn
VB_DATA[15..0]
VB_D_LTn
RD_OEn
LOCAL_DTAn
MODSELn
D0ENn
D1ENn
Note: Grounded input traces accessible on solder side
Communications Multidev JDL
1040, boulevard St-Joseph bureau 8,
Hull, Qc
J8Z 1X9
REAL TIME
TIB : VMEBUS INTERFACE
Document Number
MX002-01
Rev
A.0
Wednesday, January 17, 1996
Sheet 5 of 7

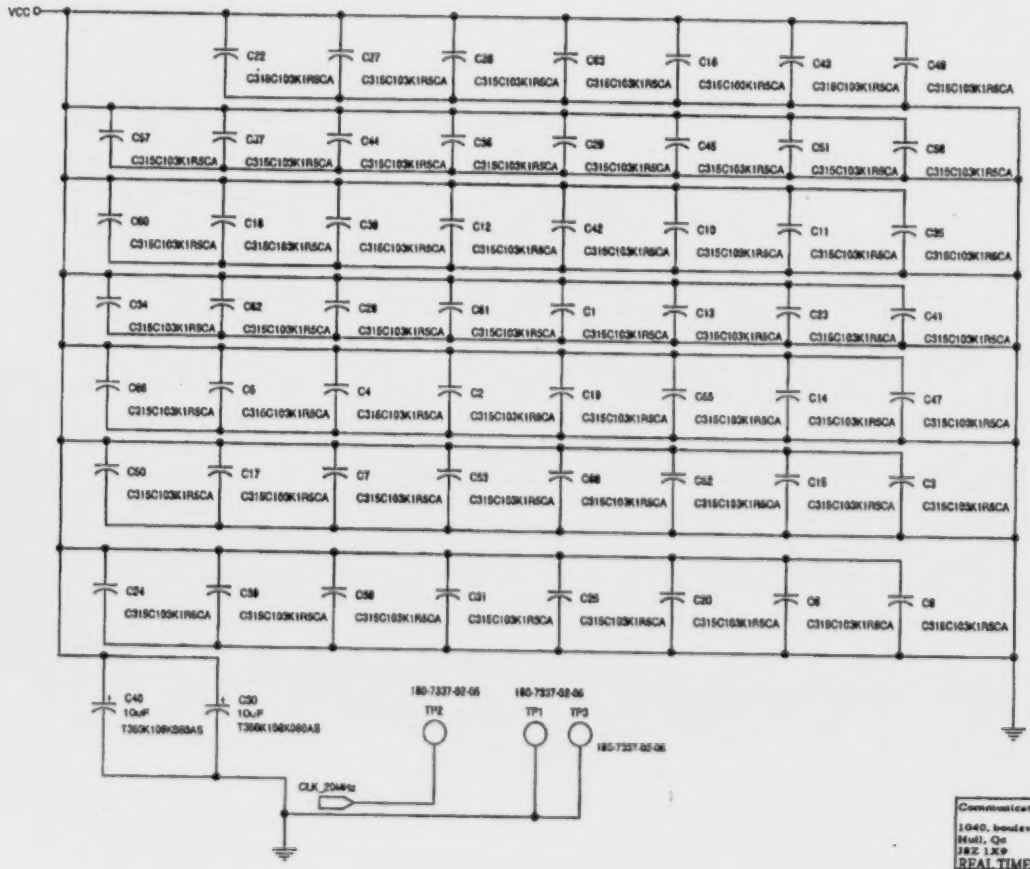


Communications Multidev JDL
1040, Boulevard St-Joseph bureau 8,
Hull, Qc
J8Z 1X9
REAL TIME

TIB: DFB AND BUFFER CONTROLS AND LED CONTROLS

Size	Document Number	Rev
B	MX002-01	A.9

Date: Wednesday, January 17, 1996 Sheet 6 of 7

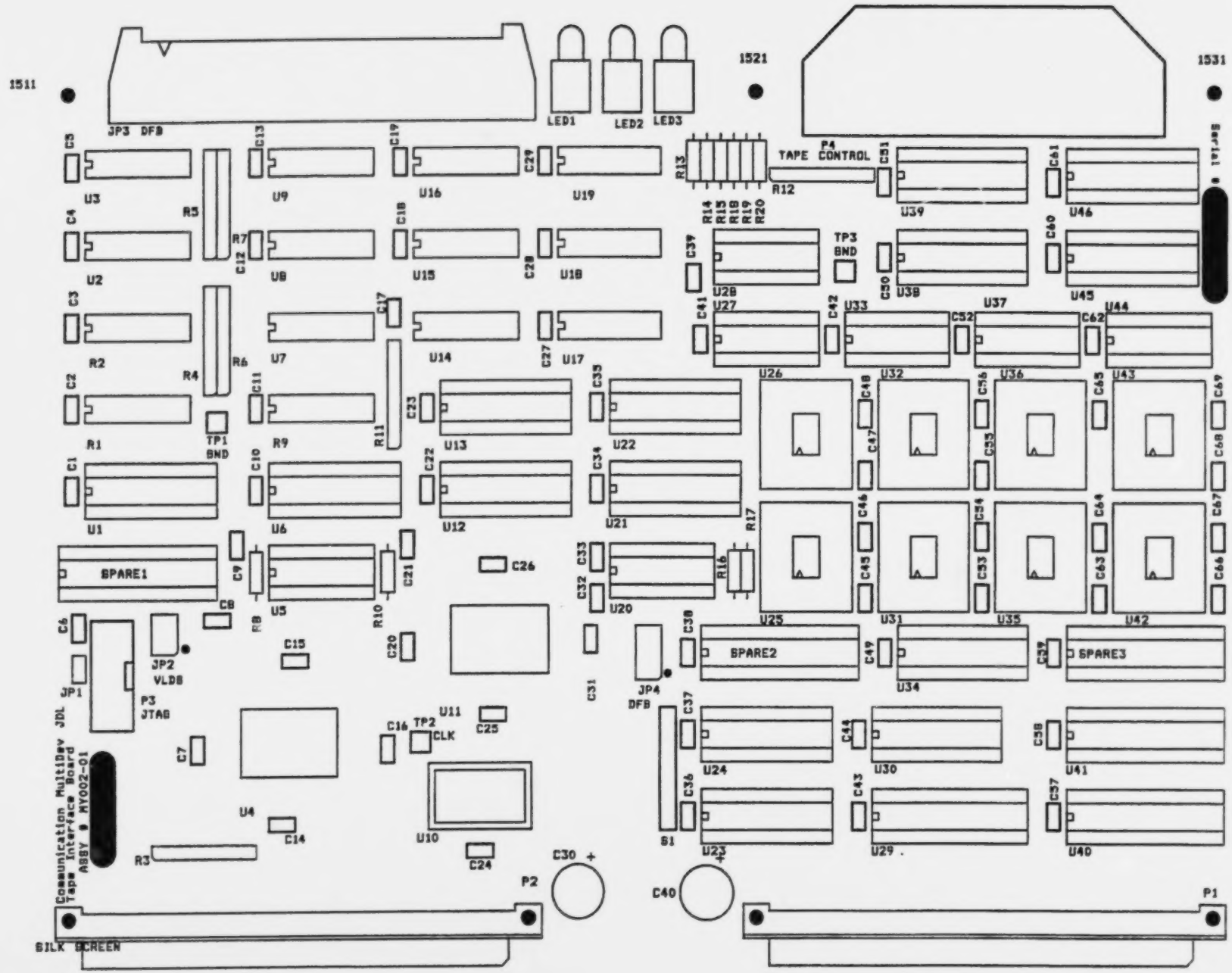


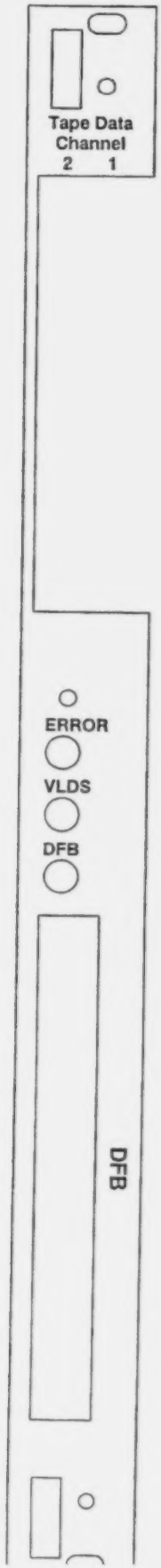
Communications Multidev JDL
1040, boulevard St-Joseph bureau 8,
Hull, Qc
J8Z 1X9
REAL TIME

DECOUPLING

Size B | Document Number MIX003-01 | Rev A.0

Date: Wednesday, January 17, 1996 Sheet 7 of 7

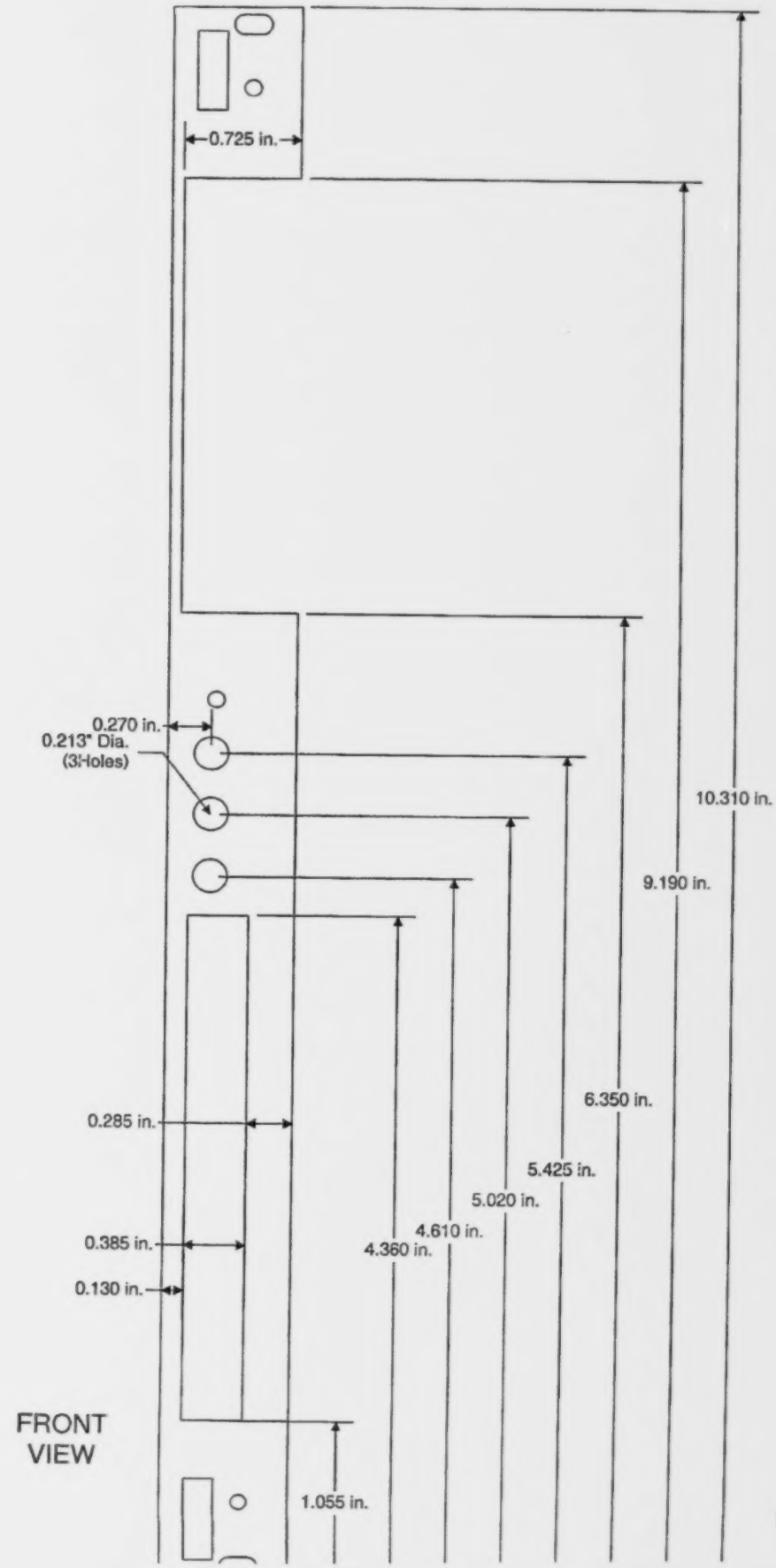

1511
1521
1531
JP3 DFB
LED1
LED2
LED3
P4
TAPE CONTROL
TP3
GND
TP1
GND
TP2
CLK
JP1
JDL
JP2
VLDB
P3
JTAG
JP4
DFB
SPARE1
SPARE2
SPARE3
S1
P1
P2
Communication Multiplex
Tape Interface Board
ASSY # MY002-01
SILK SCREEN



FRONT VIEW
FOR LABELS

TIB-VME
BRACKET
BKT002-01
REV A.0
PAGE 1 OF 2

Note: Measurements ar



TIB-VME
BRACKET
BKT002-01
REV A.0
PAGE 2 OF 2

Note: Measurements a
in Inches

APPENDIX C: INTERCONNECT

1) DFB Port
2) VMEbus P1
3) Tape Control Port
4) Tape Channel 1 Data Port
5) Tape Channel 2 Data Port

C.1 DFB PORT

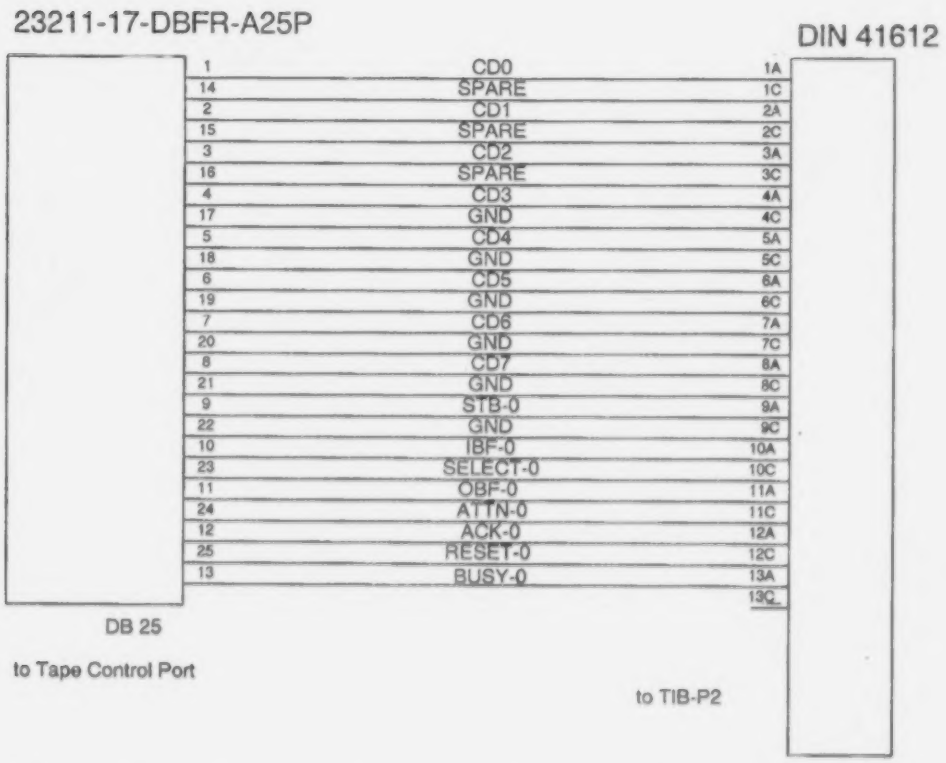
Please Refer to the Differential Fast Bus Interface Manual [3].

C.2 VMEBUS P1

Table 53
J1/P1 Pin Assignments

Pin Number	(a) Signal Mnemonic	(b) Signal Mnemonic	(c) Signal Mnemonic
1	D00	BBSY*	D08
2	D01	BCLR*	D09
3	D02	ACFAIL*	D10
4	D03	BG0IN*	D11
5	D04	BG0OUT*	D12
6	D05	BG1IN*	D13
7	D06	BG1OUT*	D14
8	D07	BG2IN*	D15
9	GND	BG2OUT*	GND
10	SYSCLK	BG3IN*	SYSFAIL*
11	GND	BG3OUT*	BERR*
12	DS1*	BR0*	SYSRESET*
13	DS0*	BR1*	LWORD*
14	WRITE*	BR2*	AM5
15	GND	BR3*	A23
16	DTACK*	AM0	A22
17	GND	AM1	A21
18	AS*	AM2	A20
19	GND	AM3	A19
20	IACK*	GND	A18
21	IACKIN*	SERCLK (1)	A17
22	IACKOUT*	SERDAT* (1)	A16
23	AM4	GND	A15
24	A07	IRQ7*	A14
25	A06	IRQ6*	A13
26	A05	IRQ5*	A12
27	A04	IRQ4*	A11
28	A03	IRQ3*	A10
29	A02	IRQ2*	A09
30	A01	IRQ1*	A08
31	-12 V	+5 V STDBY	+12 V
32	+5 V	+5 V	+5 V

C.3 TAPE CONTROL PORT

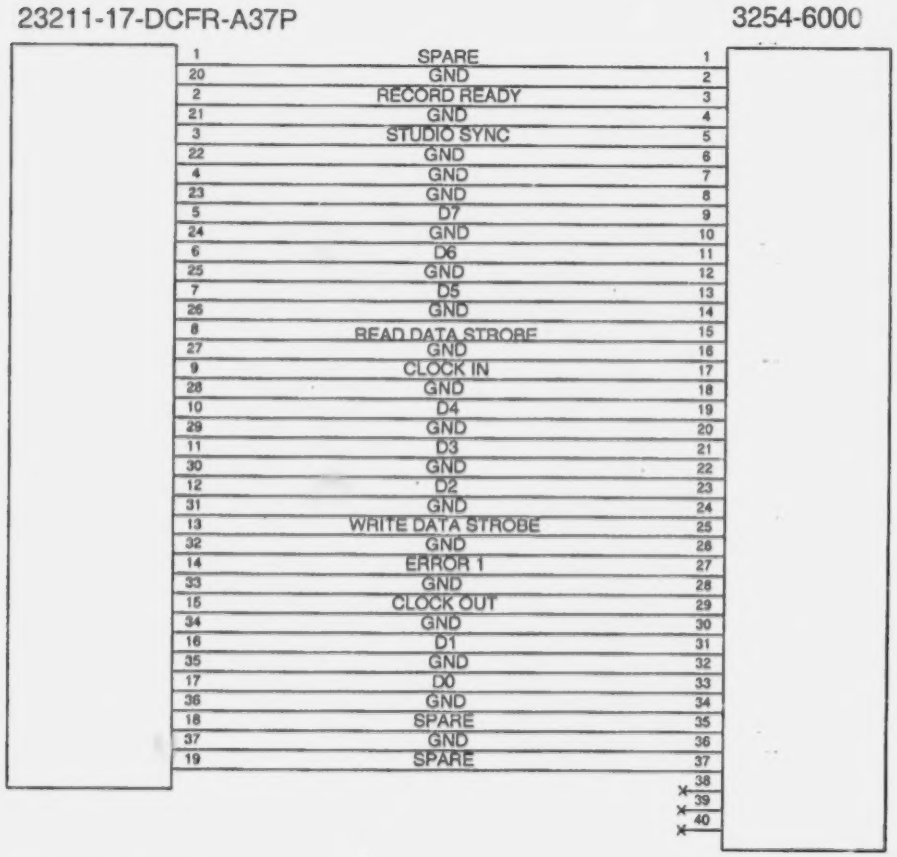


Metrum Tape Control: Tape Interface Board P2 to Back Panel Extension Cable

* Refer to page 2-8 of the VLDS Operating Manual [4]

C.4 TAPE CHANNEL 1 DATA PORT

CHANNEL 1 DATA



Metrum Tape Data: Tape Interface Board Front Panel to System Back Panel, Channel 1 Data Cable

* Refer to page 2-9 of the VLDS Operating Manual [4]

C.5 TAPE CHANNEL 2 DATA PORT

CHANNEL 2 DATA

23211-17-DCFR-A37P			3254-6000
	1	DIRECT CHANNEL IN	1
	20	DIRECT CHANNEL IN RTN	2
	2	DIRECT CHANNEL OUT	3
	21	DIRECT CHANNEL OUT RTN	4
	3	SPARE	5
	22	SPARE	6
	4	SPARE	7
	23	SPARE	8
	5	D15	9
	24	GND	10
	6	D14	11
	25	GND	12
	7	D13	13
	26	GND	14
	8	E - E STROBE	15
	27	GND	16
	9	SPARE	17
	28	SPARE	18
	10	D12	19
	29	GND	20
	11	D11	21
	30	GND	22
	12	D10	23
	31	GND	24
	13	SPARE	25
	32	SPARE	26
	14	ERROR 2	27
	33	GND	28
	15	SPARE	29
	34	SPARE	30
	16	D9	31
	35	GND	32
	17	D8	33
	36	GND	34
	18	SPARE	35
	37	SPARE	36
	19	SPARE	37
			38
			39
			40

to Tape Channel 2 Data Port

to TIB Channel 2 Data Port

Metrum Tape Data: Tape Interface Board Front Panel to System Back Panel, Channel 2 Data Cable

* Refer to page 2-9 of the VLDS Operating Manual [4]

APPENDIX D: BOARD MODIFICATIONS

The modifications that were required on the current revision of the PCB blank are:

1) To permit access of the card in both supervisory and non-privileged (VMEbus) modes, a modification was brought to U29, pin 2. On the schematics, this pin is tied to VCC. The CCA has been modified to disconnect the pin from VCC and reconnect it to GND, by first bending up pin 2 and then jumpering it, with a small wire (30 AWG) to pin 12.

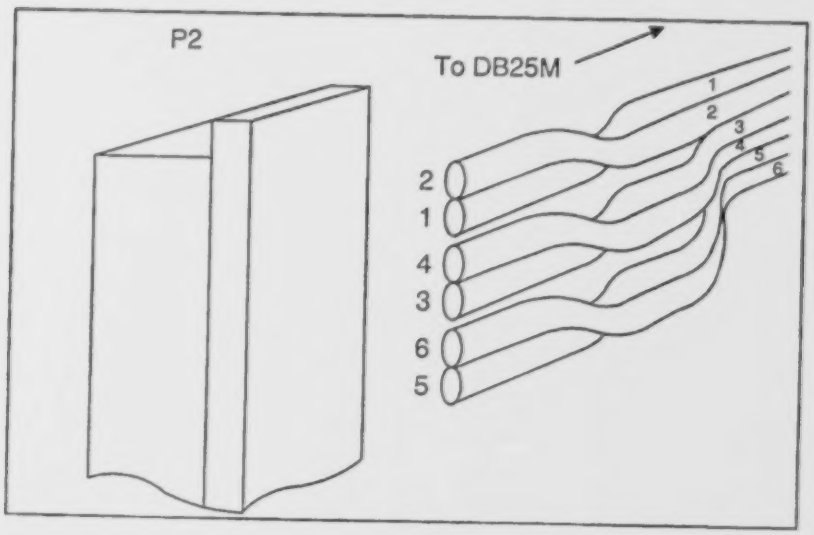
2) The following items were not installed on the CCA, because they were designed in case of termination requirements. Since no problems were discovered, and it is felt that installing them would unbalance the termination, they were not installed.

R1, R2, R4, R5, R6, R7, R9, R11, R12

3) The next items were not installed for the same reasons but this time they apply to the FIFO buffer. The capacitors were implemented to allow the insertion of decoupling in the XI/XO communication between two cascaded FIFO's. The capacitor is used to reduce ground noise on the FIFO signals, if such a case exists. In this application, no problems were found so no capacitors were required.

C46, C48, C54, C56, C64, C65, C67, C69

4) The original intent for the construction of the Tape Interface Board was to be able to fabricate all required cabling using ribbon cables and connectors. However, due to a small manufacturing defect, the fabrication of the P2 extension cable requires a special assembly.
If a ribbon cable is used, make sure that each pair is inverted before crimping the cable to the connector (i.e. Pin 1 of the DB-25 goes to Pin 2 of P2 and Pin 2 of the DB-25 goes to Pin 1 of P2, and so on). A simple solution is to take the ribbon cable on the P2 end and separate the cable every two wire. Next, by giving each pair of wire a half turn, they can be installed on the connector before crimping



UNCLASSIFIED

SECURITY CLASSIFICATION OF FORM
(highest classification of Title, Abstract, Keywords)

DOCUMENT CONTROL DATA

(Security classification of title, body of abstract and indexing annotation must be entered when the overall document is classified)

1. ORIGINATOR (the name and address of the organization preparing the document. Organizations for whom the document was prepared, e.g. Establishment sponsoring a contractor's report, or tasking agency, are entered in section 8.)

MultiDev JDL Communications Inc.
1040 St.Joseph Blvd., suite 8
Hull, Quebec J8Z 1X9

2. SECURITY CLASSIFICATION (overall security classification of the document, including special warning terms if applicable)

UNCLASSIFIED

3. TITLE (the complete document title as indicated on the title page. Its classification should be indicated by the appropriate abbreviation (S, C or U) in parentheses after the title.)

Tape Interface Board - Maintenance Manual (U)

4. AUTHORS (Last name, first name, middle initial)

Da Sylva, Pierre

5. DATE OF PUBLICATION (month and year of publication of document)

January 1996

6a. NO. OF PAGES (total containing information. Include Annexes, Appendices, etc.)

42 pages

6b. NO. OF REFS (total cited in document)

4

7. DESCRIPTIVE NOTES (the category of the document, e.g. technical report, technical note or memorandum. If appropriate, enter the type of report, e.g. interim, progress, summary, annual or final. Give the inclusive dates when a specific reporting period is covered.)

Technical Report, Operation Manual

8. SPONSORING ACTIVITY (the name of the department project office or laboratory sponsoring the research and development. Include the address.)

Defence Research Establishment Ottawa, Radar Division
3701 Carling Ave., Ottawa, Ontario

9a. PROJECT OR GRANT NO. (if appropriate, the applicable research and development project or grant number under which the document was written. Please specify whether project or grant.)

9b. CONTRACT NO. (if appropriate, the applicable number under which the document was written.)

W7714-4-9858

10a. ORIGINATOR'S DOCUMENT NUMBER (the official document number by which the document is identified by the originating activity. This number must be unique to this document.)

MD 002-06

10b. OTHER DOCUMENT NOS. (Any other numbers which may be assigned this document either by the originator or by the sponsor.)

DREO CONTRACTOR REPORT 98-635

11. DOCUMENT AVAILABILITY (any limitations on further dissemination of the document, other than those imposed by security classification.)

- (✓) Unlimited distribution
- ( ) Distribution limited to defence departments and defence contractors: further distribution only as approved
- ( ) Distribution limited to defence departments and Canadian defence contractors: further distribution only as approved
- ( ) Distribution limited to government departments and agencies: further distribution only as approved
- ( ) Distribution limited to defence departments: further distribution only as approved
- ( ) Other (please specify):

12. DOCUMENT ANNOUNCEMENT (any limitation to the bibliographic announcement of this document. This will normally correspond to the Document Availability (11). However, where further distribution (beyond the audience specified in 11) is possible, a wider announcement audience may be selected.)

Unlimited distribution

UNCLASSIFIED

SECURITY CLASSIFICATION OF FORM

13. ABSTRACT (a brief and factual summary of the document. It may also appear elsewhere in the body of the document itself. It is highly desirable that the abstract of classified documents be unclassified. Each paragraph of the abstract shall begin with an indication of the security classification of the information in the paragraph (unless the document itself is unclassified) represented as (S), (C), or (U). It is not necessary to include here abstracts in both official languages unless the text is bilingual.)

The EARS architecture allows the transfer of massive amount of information between the data sources and the storage subsystems. It is also capable of monitoring the transfer activity, set up the radar system for operation and control, and perform some data processing on the radar data.

The Data Conversion System unit is controlled by a custom designed Tape Interface Board (TIB) which initiates and manages the data transfers between the tape unit and the disk array. The TIB is also capable of transferring the information back (from the tape unit) to the disk array. The other modes of operation include the transfer of information stored on disk or tape to any devices on the VMEbus and the transfer from VMEbus to tape.

The Maintenance Manual contains the information required to ensure the proper installation and operation of the Tape Interface Board. The operator will conduct a series of steps that, if completed successfully, will indicate a properly functioning board.

14. KEYWORDS, DESCRIPTORS or IDENTIFIERS (technically meaningful terms or short phrases that characterize a document and could be helpful in cataloguing the document. They should be selected so that no security classification is required. Identifiers such as equipment model designation, trade name, military project code name, geographic location may also be included. If possible keywords should be selected from a published thesaurus. e.g. Thesaurus of Engineering and Scientific Terms (TEST) and that thesaurus-identified. If it is not possible to select indexing terms which are Unclassified, the classification of each should be indicated as with the title.)

#509862

EARS, Experimental Array Radar System
DATA ACQUISITION
REAL TIME PROCESSING
TAPE BACK-UP UNIT
TAPE INTERFACE BOARD (TIB)
VERY LARGE DATA STORAGE
VMEBUS
OPERATING SYSTEM (OS-9)
SUN WORKSTATION
METRUM TAPE UNIT
CONCEPT 51 DISK ARRAY
SYSTEM CONTROLLER
DIGITAL SIGNAL PROCESSING
GRAPHICAL USER INTERFACE

90-635

Contract #W7714-4-9858

(document MD002-04)

TAPE INTERFACE BOARD
DETAILED DESIGN DOCUMENT

written by:

Pierre Da Sylva
Pierre Blanchet

MultiDev JDL Communications Inc.
241 Dome Ave, suite 104
Hull, Quebec J8Z 3G6

January 18, 1996

REVISION SHEET

REVISION	DATE	AUTHOR
MD002-04	January 18, 1996	Pierre Da Sylva, Pierre Blanchet
Approval: Jean Caseault		

GLOSSARY

CPU:	*Central Processing Unit.*
DFB:	*Differential Fast Bus*, used by the disk array.
DLCS:	*Data Logger and Control System.* The VME-based data acquisition system.
DOS:	*Disk Operating System,* from Microsoft.
EARS:	*Experimental Array Radar System.*
FIFO:	*First-in, First-out* memory.
IF:	*Intermediate Frequency*
I/O:	*Input/Output*
PC:	*Personal Computer.*
SCSI:	*Small Computer System Interface*
TIB:	*Tape Interface Board.*
VLDS:	*Very Large Data Storage.*

TABLE OF CONTENTS

1. INTRODUCTION 7
1.1 DOCUMENT SCOPE 7
1.2 THEORY OF OPERATION 7
2. FUNCTIONAL DESCRIPTION 9
3. DETAILED DESCRIPTION 10
3.1 DISK ARRAY INTERFACE 10
3.2 TAPE INTERFACE 10
3.2.1 Control Port 10
3.2.2 Data Port 12
3.3 TRANSFER BUFFERS 12
3.4 LOGIC CONTROL BLOCKS 12
3.4.1 VLDS Controls 13
3.4.2 DFB Controls 13
3.4.3 FIFO Controls 14
3.4.4 VMEbus Controls 14
3.5 VMEBUS INTERFACE 14
3.5.1 VMEbus Data Port 14
3.5.2 Control Register 14
3.5.3 Tape Command Port 17
3.5.4 FIFO Reset Register 17
3.5.5 Tape Interface Board Reset Register 17
4. TRANSFER PROCESS 18
4.1 INITIALIZATION 18
4.1.1 Power Up Initialization 18
4.1.2 Buffer Initialization 18
4.1.3 Board Initialization 18
4.2 TRANSFER OPERATIONS 19
4.2.1 Data Direction 19
4.2.2 Disk/Tape Preparation 21
4.2.3 Differential Fast Bus Initialization 21
4.2.4 VLDS Initialization 21
5. PROGRAMMING MODEL 22
5.1 VMEBUS ACCESS 22
5.1.1 VMEbus Data Port (x80000 to xFFFFF) 23
5.1.2 Control Register (x00000) 23
5.1.3 Tape Command Port (x00002) 27
5.1.4 FIFO Reset Register (x00004) 27
5.1.5 Tape Interface Board Reset Register (x00006) 27

6. SOFTWARE DESCRIPTION 29
6.1 TIB AND METRUM RESET 29
6.2 TIB INITIALIZATION 30
6.3 METRUM INITIALIZATION 30
6.4 FORMAT A CASSETTE 30
6.5 SKIP FILEMARKS ON A METRUM CASSETTE 31
6.6 REWIND A METRUM CASSETTE 31
6.7 EJECT A METRUM CASSETTE 32
6.8 TRANSFER FROM DISK ARRAY TO METRUM 32
6.9 TRANSFER FROM METRUM TO DISK ARRAY 35
6.10 TRANSFER FROM METRUM TO EXABYTE 38
6.11 VERIFY THE PRESENCE OF A CASSETTE 41
6.12 CHECK IF THE CASSETTE IS WRITABLE 41
6.13 SEND COMMANDS TO THE METRUM 41
6.14 READ STATUS FROM THE METRUM 41
7. REFERENCES 43
8. CONNECTOR PIN-OUT 44
8.1 TAPE DATA PORT 44
8.2 TAPE CONTROL PORT 45
8.3 DIFFERENTIAL FAST BUS 46
8.4 VMEBUS CONNECTORS 46

LIST OF FIGURES

Figure 1-1: EARS Data Acquisition and Control System 8
Figure 2-1: TIB Operating Modes 9
Figure 3-1: TIB Block Diagram 11
Figure 4-1: Disk to Tape Transfer Sequence 19
Figure 4-2: Tape to Disk Transfer Sequence 20
Figure 5-1: VMEbus Data Port 23
Figure 5-2: Control Register Bit Assignment 23
Figure 5-3: Tape Command Port 27
Figure 5-4: FIFO Reset Register 27
Figure 5-5: TIB Reset Register 27
Figure 6-1: Disk Array to Metrum Transfer Algorithm 34
Figure 6-2: Metrum to Disk Array Transfer Algorithm 37
Figure 6-3: Metrum to Exabyte Transfer Algorithm 40

LIST OF TABLES

Table 5-1: SIP Switch Assignment 22
Table 5-2: VMEbus Register Map 22
Table 5-3: Data Direction Control Bits 24
Table 5-4: Error Bit Assignment 24
Table 5-5: FIFO Status Bit Assignment 25
Table 5-6: Start Transfer Bit 26
Table 5-7: DFB Master Enable Bit 26
Table 5-8: TIB Reset Register Bit Definition 28

1. INTRODUCTION

1.1 DOCUMENT SCOPE

This document describes the architecture of the Tape Interface Board to be used in the Data Logger and Control System (DLCS) of the Experimental Array Radar System (EARS) project. The document includes a high-level description of the system operation and the hardware components. The Tape Interface Board has been developed for the EARS project, as required in Task 9 of contract W7714-4-9858. It has been designed according to the Project Plan and MultiDev's proposal, reference [1].

1.2 THEORY OF OPERATION

The present configuration of the EARS data acquisition and control system is shown in Figure 1-1.

The EARS architecture allows the transfer of massive amount of information between the data sources (IF outputs of the radar vertical and horizontal arrays) and the storage subsystems (hard disk array, Metrum tape unit and Exabyte tape unit). It is also capable of monitoring the transfer activity, set up the radar system for operation and control, and perform some data processing on the radar data.

During a data acquisition session, the Data conversion System sends the radar data to the Disk Array via the Differential Fast Bus (DFB). Once the information is stored on disk at full speed (20 MB/sec), the acquired information must be backed-up on tape to free the disk array for another session. A Metrum tape unit is used to perform tape back-up at high speed (4 MB/sec). This unit is controlled by a custom designed Tape Interface Board (TIB) which will initiate and manage the data transfers between the tape unit and the disk array. The TIB will also be able to transfer the information back (from the tape unit) to the disk array. The other modes of operation include the transfer of information stored on disk or tape to any devices on the VMEbus and the transfer from VMEbus to tape.

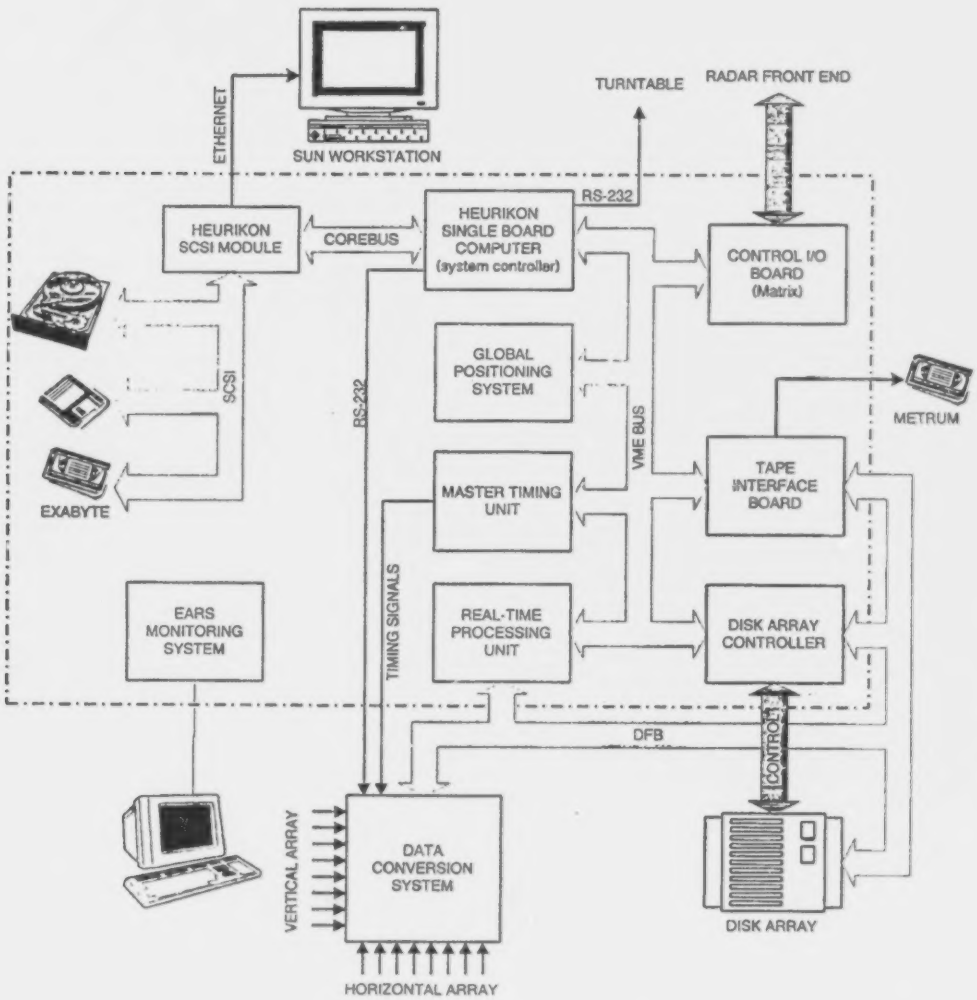


Figure 1-1: EARS Data Acquisition and Control System

2. FUNCTIONAL DESCRIPTION

The TIB is designed to allow bi-directional transfers between the tape and disk units. Five modes of operation are supported (see Figure 2-1). In the first mode, the card receives data from the disk array at a rate of 10 MHz (20 MB/sec, 16-bit words) and sends it to the tape at a rate of 2 MHz (4 MB/sec, 8-bit words, 2 channels). For the second mode of operation, the tape sends its data to the Tape Interface Board, where it is returned to the disk array. The data is temporarily buffered into the Tape Interface Board to balance the transfer rates. Two additional modes are implemented to monitor the disk array data or the tape data via VMEbus. The Tape Interface Bus acts as a large buffer during those transfers. Finally, the last mode allows the VMEbus to write a data block to the tape for testing purposes.

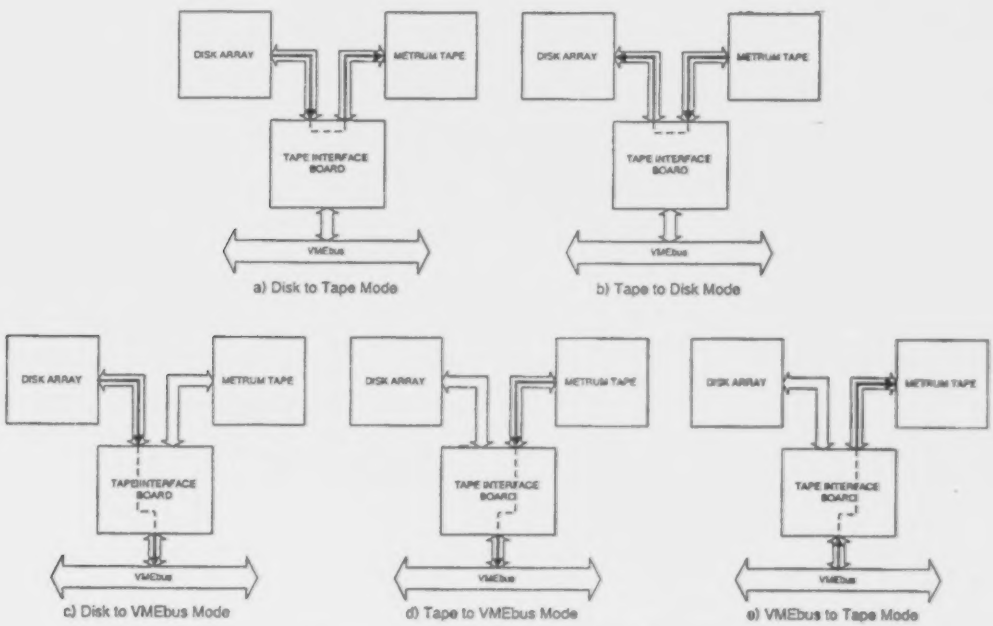


Figure 2-1: TIB Operating Modes

The board is designed to conform to the VMEbus standard, the DFB standard as defined by Storage Concept (for the disk array) [2], and the data and control ports of the Metrum tape unit [3].

3. DETAILED DESCRIPTION

The following section describes in details the operation of the different sections of the Tape Interface Board:

- the disk array interface, via the DFB (Differential Fast Bus),
- the tape interface (Metrum's VLDS standard interface), data and controls,
- transfer buffers,
- the logic control for the tape interface, the disk interface and the buffers, and
- the VMEbus interface.

3.1 DISK ARRAY INTERFACE

The disk array interface implements the DFB standard as defined by Storage Concept in reference [2]. The interface recognizes signals from the Differential Fast Bus control lines and informs the main control logic of the state of the Differential Fast Bus.

Figure 3-1 shows a block diagram of the Tape Interface Board.

The DFB control logic provides all the signalling required to perform the following operations:

- manage the control signals on the DFB for master mode operation.
- manage the write cycles from a source device on the DFB to the buffer memory on the Tape Interface Board. The source device is the disk array.
- manage the read cycles from the buffer memory on the TIB to a destination device on the DFB. The destination device is the disk array.

After setting the direction (or operation mode) and enabling the Differential Fast Bus mastership bit in the control register (section 3.5.2.5), the disk array transfer can be initiated via the Disk Array Controller. From that point on, the process is automatic (see reference [2]).

3.2 TAPE INTERFACE

The tape interface contains two sections: the control port and the data port.

3.2.1 Control Port

The control section (VLDS CONTROL) is used to transmit commands to the tape unit. Commands are issued to the tape unit via the VMEbus (a memory mapped Tape Command Register is accessible over the VMEbus). Status registers on the tape unit are read through the same register. For a complete list of commands and status words, see sections 3-7.4, 3-12.10 and 3-17 of the Metrum Technical Manual [3].

Note: The programmer must ensure proper data formatting for all tape commands. Refer to the Metrum Technical Manual for the basic VLDS command formats.

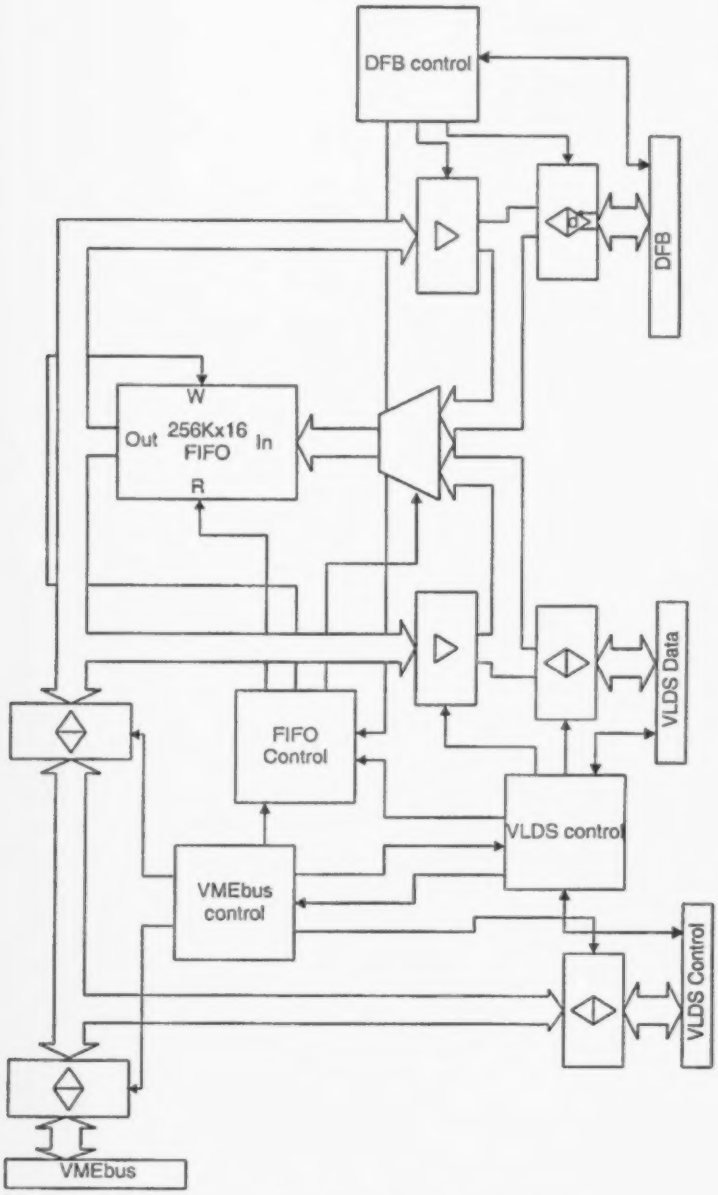


Figure 3-1: TIB Block Diagram

All VLDS and VMEbus controls are managed by the VLDS control and VMEbus control blocks respectively (see Figure 3-1).

3.2.2 Data Port

The Data Port transfers data to and from the tape unit. This port is a proprietary high speed port (4 MB/sec) defined by Metrum (please refer to the Metrum User's Manual for the timing and the protocol). The VLDS control block manages the handshake signals to the tape unit. It also communicates with the FIFO controller for data manipulation.

3.3 TRANSFER BUFFERS

The transfer buffers temporarily store the tape or disk data during the transfers. The TIB Control Register (section 5.1.2) determines the direction of the transfer. In the normal modes of operation (modes 1 and 2), transfers can be from disk to tape or from tape to disk. When data monitoring is required at a low level, the data buffers can be read via the VMEbus (modes 3 and 4) or written by the VMEbus (mode 5).

The FIFO Control block monitors the tape and disk interface signals to generate proper FIFO Write and Read signals. The control register dictates the direction of data to this control block.

3.4 LOGIC CONTROL BLOCKS

Each TIB process is accomplished by one or more control blocks. There are four control blocks for the card. They are:

- The DFB control block
- The VLDS control block
- The FIFO control block
- The VMEbus control block

The control blocks are implemented using 2 high density programmable gate arrays. The VMEbus and VLDS controls are implemented by one PGA and the DFB and FIFO controls by the other.

For disk to tape transfers, the FIFO control block enables the data multiplexer between the FIFO inputs and the disk interface connector, and the data multiplexer between the FIFO outputs and the tape interface connector. Signals from the DFB control block are sent to the FIFO control block to implement FIFO writes. FIFO reads are initiated by the VLDS control block. Data output buffers to the disk interface and data input buffers from the tape interface are disabled (by the direction control register from the VMEbus Control block).

For tape to disk transfers, data buffers are enabled between the FIFO inputs and the tape interface connector, and between the FIFO outputs and the disk interface connector. Signals from the VLDS control block are sent to the FIFO control block to implement FIFO writes. FIFO reads are controlled by the DFB control block. Data output buffers to the tape unit and data input buffers from the disk interface are disabled (by the direction control register from the VMEbus Control block).

For disk to VME transfers, data buffers are enabled between the FIFO inputs and the disk interface connector, and between the FIFO outputs and the VMEbus connector. Signals from the DFB control block are sent to the FIFO control block to implement FIFO writes. FIFO reads are controlled by the VMEbus control block. Data output buffers to the disk interface and data input buffers from VMEbus are disabled (by the direction control register from the VMEbus Control block).

For tape to VME transfers, data buffers are enabled between the FIFO inputs and the tape interface connector, and between the FIFO outputs and the VMEbus connector. Signals from the VLDS control block are sent to the FIFO control block to implement FIFO writes. FIFO reads are controlled by the VMEbus control block. Data output buffers to the tape unit and data input buffers from VMEbus are disabled (by the direction control register from the VMEbus Control block).

For VME to tape transfers, a two step process is required. First, data buffers are enabled between the FIFO inputs and the VMEbus connector. Signals from the VMEbus control block are sent to the FIFO control block to implement FIFO writes for a complete data block. Afterwards, data buffers are enabled between the FIFO outputs and the tape interface connector. FIFO reads are controlled by the VLDS control block. Data output buffers to VMEbus and data input buffers from the tape unit are disabled (by the direction control register from the VMEbus Control block).

VMEbus operation is controlled by the VMEbus control block. Tape controls are provided by the system controller via the VMEbus.

3.4.1 VLDS Controls

The VLDS control block monitors signals from the tape interface and, as regimented by the Control Register, issues proper data buffer controls and FIFO control block signals.

The tape interface requires two ports to be controlled: the data port and the command port. The data port requires automatic handshake controls and is used exclusively for data transfers. Controls are issued directly from the VLDS control block.

The command port is used to control tape operations. The commands are issued by the system controller on the VMEbus. The VLDS control block receives signals from the VMEbus control block and performs protocol exchanges following the Metrum specifications for the command port.

For the tape Status Read command, the VMEbus host performs a Tape Command Port Register read (see section 3.5.3) to obtain an 8-bit Tape Status word. The VLDS control block generates proper signal protocol for this operation, by managing the handshaking signals between the VMEbus control block and the tape command port.

3.4.2 DFB Controls

The DFB control block monitors signals from the DFB connector and, as regimented by the Control Register, issues proper data buffer controls and FIFO control block signals.

The disk interface receives DFB signals and performs proper handshaking. The interface behaves as a master on the Differential Fast Bus as dictated by the Differential Fast Bus Interface Manual (rev. 2.0) from Storage Concepts [2]. As data is being transferred, FIFO qualifying signals are generated for the FIFO control block. This process is fully automatic and requires no software control.

3.4.3 FIFO Controls

The FIFO control block monitors signals from the other three control blocks and generates FIFO write and read control signals accordingly. Signals coming from the VMEbus control block determine the proper sequence of these signals.

3.4.4 VMEbus Controls

The VMEbus interface uses the PLX VME2000 (VME interface, slave mode) and the VMEbus control block to perform all VMEbus functions. The VMEbus control block is responsible for command transfers to the tape unit and selecting one of five data transfer modes available. Data direction is controlled by the VMEbus control block. Signals from this block are sent to instruct the FIFO control block on proper FIFO signal generation. Status of the operations are monitored via this block.

3.5 VMEbus INTERFACE

The VMEbus interface allows control of the Tape Interface Board. The VMEbus interface permits the sending of commands from the system controller to the tape unit. The VMEbus interface also selects the data transfer mode. Data transfers modes 3, 4 and 5 (involving VMEbus) allow the monitoring of the disk array, the tape unit and the FIFO memories. The VMEbus handshakes to A24D16 transfer type only. Any other types of accesses are ignored (VMEbus time-out). The board supports single word for register access and single or block transfers for the data port access.

Address selection on the VMEbus is switch-selectable. The card decodes a block of 1 Mbytes on VMEbus (x00000 to xFFFFF).

3.5.1 VMEbus Data Port

The VMEbus data port is used to read the FIFO memories during data monitoring. By setting the mode bits properly, either the tape data or the disk data can be monitored. The size of the data block to be monitored is fixed by the size of the FIFO buffers (128K x 16, expandable to 256K x 16). Data is made available on a 1 Mbyte memory allocation. It supports block transfers (BLT) on VMEbus. This data port can also write to FIFO for testing the tape interface.

3.5.2 Control Register

The Control Register provides control and information about the behaviour of the card. Some bits are used to select one of the five data transfer modes which will take place on

the card. Another bit is used to perform transfer initialisation. When set, the interface specified by the mode control bits initiates transfer procedures.

Five error flags are also provided to indicate improper transfer performance when VMEbus monitoring is accomplished on the FIFO's. Reading the data direction bits provides information on the direction of data transfers.

3.5.2.1 Data Direction

The data direction bits provide control and monitoring of the data transfers accomplished by the Tape Interface Board. They are set via VMEbus and can be read at any time to inform on the direction of the transfers.

Five modes are supported:

- Disk to tape transfers
- Tape to disk transfers
- VMEbus monitoring of disk data
- VMEbus monitoring of tape data
- VMEbus to tape transfers

For disk to tape transfers, the Differential Fast Bus is enabled (all other Differential Fast Bus controllers shall be off-line) and operates in master mode (Continuous) with Slave (disk array) to Master (TIB) Transfers. The TIB monitors the Differential Fast Bus data lines and transfers the data to the FIFO. The FIFO output buffers are enabled onto the tape Data Port. The VLDS interface is operating in Record mode.

For tape to disk transfers, the VLDS interface operates in Reproduce mode. The TIB monitors the tape Data Port and sends the data to the FIFO's. The FIFO output buffers are driving data onto the Differential Fast Bus data lines. The Differential Fast Bus is configured to perform Master (TIB) to Slave (disk array) transfers.

For VMEbus monitoring modes, two sources of data can be selected. For disk monitoring, the FIFO data buffers are capturing data from the Differential Fast Bus data lines. For tape monitoring, the FIFO data buffers are capturing the tape Data Port data lines.

For VMEbus to tape transfers, the VMEbus is enabled and writes a block to the FIFOs. Only after the write cycle is complete, the FIFO output buffers are enabled onto the tape Data Port for a complete read cycle. The VLDS interface is operating in Record mode.

3.5.2.2 Error Detection

Error detection is provided to indicate improper transfers. These conditions are detected in various locations depending on their effect on the system.

Metrum Data Transfer Block Error - One source of error is the detection of semaphores from the Metrum data interface (Error-1, Error-2). The assertion of either signals indicates an error in the operation of the tape.

Metrum Tape Command Port Error - Sending a command other than an exception command while BUSY-0 is set, flags a bit in the Control Register. Monitoring the tape unit status registers should provide a more detailed view of tape problems. For more details on the operation of the Metrum tape unit, please refer to the Metrum Technical Manual for the VLDS command formats [3].

Metrum Tape Unit Not Selected - Issuing a command to the tape unit when it is not selected will cause an error flag to be set.

Disk Unit Has Wrong Transfer Mode - An error can be detected on the Differential Fast Bus if the master (the Tape Interface Board) detects an invalid state on the BLOCK signal.

Metrum Tape Attention Flag - The tape unit indicates trouble by asserting the ATTN-0 line on the tape command port.

3.5.2.3 FIFO Status

Two bits are assigned in the Control Register to monitor the buffer status flags. The FIFO_EMPTY bit indicates when both FIFO buffers are empty. This flag is asserted when a power-up or a software reset is applied to the Tape Interface Board. The second flag indicates when both buffers are full. The FIFO_FULL flag is de-asserted after a power-up or software reset.

3.5.2.4 Start Transfer

The Start Transfer bit initiates the transfer procedures. For disk to tape transfers, the disk interface indicates to the disk controls that the Tape Interface Board is ready to accept data from the disk array. For tape to disk transfers, the tape interface indicates to the tape unit that transfers are ready to be accepted from the tape.

Transfers occur until no more data needs to be read. This is determined by a time-out bit in the control register (section 5.1.2). At the end of data transfers, the Start Transfer bit is reset, indicating that more transfers can be accomplished.

In the VMEbus monitoring modes, this bit is used to initiate the transfer cycle. When the complete block has been captured, the bit is de-asserted.

3.5.2.5 Differential Fast Bus Master Enable

This bit enables the Tape Interface Board to assume control of the Differential Fast Bus. Before enabling this bit, the programmer must ensure that all other masters have been removed from the bus. After transfers are complete, the program must reset this bit to disable the Differential Fast Bus master on the Tape Interface Board.

3.5.3 Tape Command Port

The Tape Command port issues commands to the tape unit. It operates in write mode (sending data to the tape unit) except when reading a Status register on the tape unit. The port issues 8 bits (LSB) of data to the tape interface. The 8 MSB's are ignored.

When using the Status Read command, the proper data sequence must be followed (see Metrum VLDS Manual for more details). Failure to do so will result in indeterminate operation of the tape drive. Following a Status Read command, reading the Tape Command Port will return the value issued by the tape unit (8 LSB's).

3.5.4 FIFO Reset Register

The FIFO Reset register is used to reset the FIFO memories via software control. The action clears the FIFO pointers and empties the memories. The FIFO empty flag is set when a reset is given.

3.5.5 Tape Interface Board Reset Register

The Tape Interface Board Reset register is used to place the Tape Interface Board card in a known state. Following power up, it is recommended to execute this action. Once initialized, the Differential Fast Bus port is placed in an off-line mode (high impedance). The tape interface signals are taken to an idle state (de-asserted). The FIFO memories are cleared. All control register flags are reset and the control bits are placed in an unasserted state. See Section 5.1 for register default values.

If a reset is required only to a specific interface (tape or disk), controls are implemented to execute this function.

4. TRANSFER PROCESS

The following section describes the proper sequence to follow when transferring data. The process is divided in two sections: Initialization and Transfer operation.

4.1 INITIALIZATION

The initialization of the Tape Interface Board should be performed before beginning any transfers. The Tape Interface Board initialization can be divided into three sections: Power up initialization, FIFO buffer initialization and board initialization.

4.1.1 Power Up Initialization

When the system is powered up, the Tape Interface Board is automatically reset to a known, idle state. The Differential Fast Bus interface signals are placed in high impedance (DFB_ME is cleared, section 5.1.2.5). The VLDS interface is in idle mode. The FIFO's are reset (FIFO Empty Flag asserted, FIFO Full Flag is reset). The Control Register is reset with all error bits cleared and the data direction bits are monitoring the FIFO buffers (see section 5.1.2.1). Also, the Start_Transfer bit is cleared.

A board reset can be executed at any time by the use of the Tape Interface Board reset register (see section 5.1.5) under program control.

4.1.2 Buffer Initialization

The buffer can be initialized via software at any time. It is recommended that, anytime transfers are to be accomplished, the buffers be initialized. During transfers, if the buffer sequencing gets corrupted, the FIFO might need to be placed in a known state. Resetting the FIFO ensures proper operation for the transfer to come.

Buffer initialization is performed when a high level ("1") is written to the FIFO reset register on bit "D0" (see section 5.1.4 for programming model).

4.1.3 Board Initialization

Board initialization is required to ensure correct transfer procedures. A complete board initialization is accomplished by writing a high level ("1") on "D2" of the Tape Interface Board reset register (see section 5.1.5). Following this action, the card and its associated interfaces are placed in an idle state. To accomplish data transfers, the board must have all the appropriate bits set following this reset.

Individual interfaces can be reset separately if this is required. Writing a high level ("1") to the appropriate bit will reset the individual interface. Refer to section 5.1.5 for more details on resetting individual interfaces.

4.2 TRANSFER OPERATIONS

The following section describes what processes are required in order to accomplish a proper and trouble-free transfer.

4.2.1 Data Direction

All transfers require the specification of the source and destination for these transfers. The source and destination devices can be the tape, the disk or the VMEbus data port. The Control Register allows control of these options via software control.

4.2.1.1 Disk to Tape

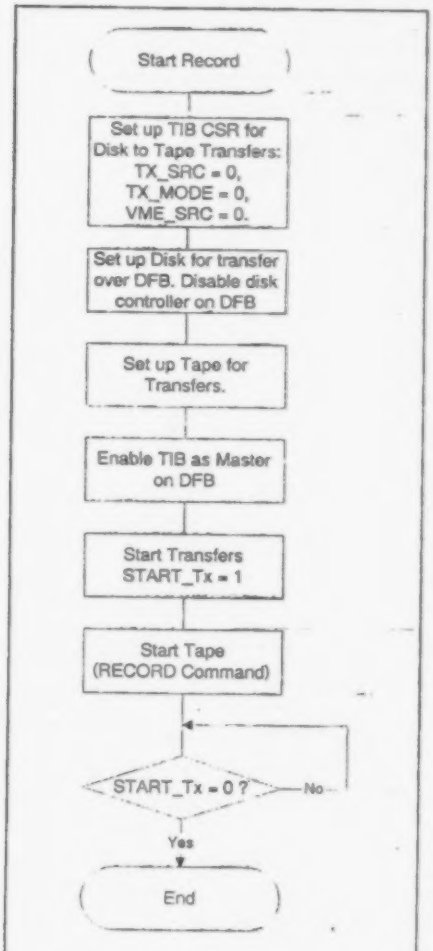

Start Record
Set up TIB CSR for Disk to Tape Transfers: TX_SRC = 0, TX_MODE = 0, VME_SRC = 0.
Set up Disk for transfer over DFB. Disable disk controller on DFB
Set up Tape for Transfers.
Enable TIB as Master on DFB
Start Transfers START_Tx = 1
Start Tape (RECORD Command)
START_Tx = 0 ?
No
Yes
End

Figure 4-1: Disk to Tape Transfer Sequence

For disk to tape transfers, the source is the disk unit, the destination, the tape unit. In the Control Register, the TX_SRC bit must be set to disk (Writing a "0" to D0 specifies the source as the disk unit). Refer to section 5.1.2.1 for a more detailed look at the Control Register. To specify the type of transfer, the TX_MODE bit is set to Disk/Tape mode ("0") and the VME_SRC bit is set to "0". Once the mode is set properly, selecting the source automatically selects the destination. To start transfers, a special sequence of operation must be followed for these transfers.

Disk to tape transfers require an eight step process. Refer to Figure 4-1 for proper sequencing of controls. The first step involves setting the mode and source controls of the Control Register. The second step requires that the disk be prepared for data transfers. Refer to the disk manuals for preparing the disk for data transfers. Following this procedure, the disk controller should be removed from the Differential Fast Bus (Differential Fast Bus master disable). The third step requires that the tape be prepared for recording.

For the fourth step, the Tape Interface Board is assigned the mastership of the DFB by setting D12 of the Control Register. The next (fifth) step involves asserting the Start_Transfer bit of the Control Register (see section 5.1.2.4). Once this is done, the disk starts issuing data transfers through the DFB.

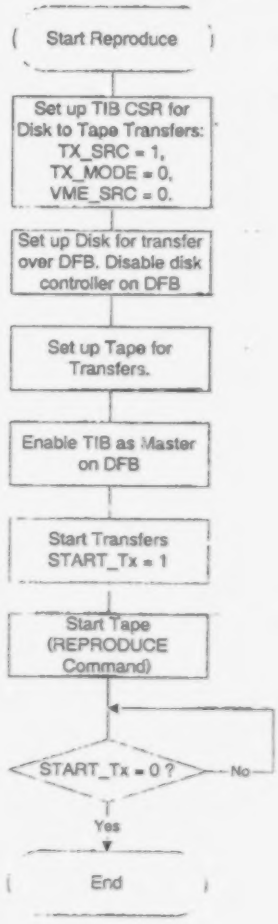

Start Reproduce
Set up TIB CSR for Disk to Tape Transfers: TX_SRC = 1, TX_MODE = 0, VME_SRC = 0.
Set up Disk for transfer over DFB. Disable disk controller on DFB
Set up Tape for Transfers.
Enable TIB as Master on DFB
Start Transfers START_Tx = 1
Start Tape (REPRODUCE Command)
START_Tx = 0 ?
No
Yes
End

Figure 4-2: Tape to Disk Transfer Sequence

At this point, the tape should be issued the Record command (51H) via the tape command port (section 5.1.2.6). Once the command is issued, the process is automated and will terminate when the START_Tx bit is reset, which can be monitored via the VMEbus.

4.2.1.2 Tape to Disk

Tape to disk transfers require a similar process as disk to tape transfers. Figure 4-2 shows the sequence to follow. In the Control Register, the TX_SRC bit must be set to tape (Writing a "1" to D0 specifies the source as the tape unit). To specify the type of transfer, the TX_MODE bit is set to Disk/Tape mode ("0") and the VME_SRC bit is set to "0". The process is Initiated by setting up the disk for transfers over the Differential Fast Bus. Once again, the mastership of the DFB is released on the disk controller. Next, the tape is initialized at the proper place for reproduction. Following this, the Start_Transfer bit is set to initiate tape to disk transfers. Once these processes are complete, the program can issue the Reproduce command (47H) to the tape. This process is automated and will terminate when the last transfer is encountered. The process can be monitored by reading the state of the Start_Transfer bit in the Control Register (D11).

4.2.1.3 VMEbus Monitoring

For monitoring either the disk or the tape through VMEbus, the unit to monitor must be properly initialized. The Control Register bits that control the data direction must be set accordingly. The TX_MODE bit is set to "1", defining the mode to VMEbus monitoring.

For monitoring the disk, the TX_SRC bit is cleared ("0"). The tape unit is monitored when the TX_SRC bit is set ("1").

Before initiating transfers through VMEbus, the source unit should be prepared accordingly. When transfers are initiated, only the first block through the specified interface (disk or tape) will be recorded in the buffer.

4.2.1.4 VMEbus to Tape

For writing a block to the tape through VMEbus, the unit must be properly initialized. The Control Register bits that control the data direction must be set accordingly. The VME_SRC bit is set to "1", defining the mode for VMEbus to Tape transfers. TX_SRC and TX_MODE are "don't care".

Before initiating transfers through VMEbus, the tape unit should be prepared accordingly. When transfers are initiated, only one block will be written into the buffer. After the block is written, the tape interface will transfer this block to the tape.

4.2.2 Disk/Tape Preparation

Before starting transfers, the source unit (disk or tape) must be prepared for transfers. For the disk unit, refer to the Disk Operating Manual [2]. For proper tape preparation, refer to the Metrum Tape Technical Manual [3].

4.2.3 Differential Fast Bus Initialization

To obtain proper operation during transfers, the disk controller must be removed from DFB mastership prior to initiating transfers. Once the hard disk controller is removed from the Differential Fast Bus, a new master can be defined. The Tape Interface Board is designed to operate as master for the transfers. To enable mastership of the Tape Interface Board on the Differential Fast Bus, the DFB_Master Enable bit of the Control Register must be set (section 5.1.2.5).

4.2.4 VLDS Initialization

The Metrum Tape unit should be initialized previous to any transfers. This ensures that the tape is at the proper place. The Metrum Technical Manual describes the required set-up. If the state of the VLDS is unknown, it is recommended that a reset be applied to the interface (see section 5.1.5).

5. PROGRAMMING MODEL

The following section describes in details the ports and registers available under program control.

5.1 VMEbus ACCESS

The VMEbus handshakes to A24D16 transfers only (AM codes 39h and 3Bh). Any other types of accesses are ignored (VMEbus time-out).

Address selection on the VMEbus is switch-selectable. The card decodes a block of 1 Mbyte on VMEbus (x00000 to xFFFFF). To select a base address, the SIP switches must be set as per Table 5-1.

Switch #	Address Bit	Compares To:	
		Open	Closed
SW1-5	A20	High	Low
SW1-6	A21	High	Low
SW1-7	A22	High	Low
SW1-8	A23	High	Low

Table 5-1: SIP Switch Assignment

The next table contains the assignment of the different registers on the VMEbus.

Address	Description	Width (Bits)
x00000	Tape Interface Board Control Register	16
x00002	Tape Command Port	8
x00004	FIFO Reset Register	1
x00006	Tape Interface Board Reset Register	5
x80000 to xFFFFF	VMEbus Data Port	16

Where "x" is the setting of the VMEbus address select SIP switches.

Table 5-2: VMEbus Register Map

5.1.1 VMEbus Data Port (x80000 to xFFFFF)

The VMEbus data port is used to read the FIFO memories for data monitoring. By setting the mode bits properly, either the tape data or the disk data can be monitored. The size of the data block to be monitored is fixed by the size of the FIFO buffers (2 sets of 128K x 16). Data is made available on a 512 Kbyte memory location which supports block transfers (BLT) on VMEbus.

D15	D14	D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0
MSB	D14	D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	LSB

Figure 5-1: VMEbus Data Port

5.1.2 Control Register (x00000)

The Control Register provides information about the behavior of the card. Special bits are set up to provide direction control of the data transfers. They allow five modes of operation to take place on the card.

D15	D14	D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0
N/U	N/U	N/U	DFB ME	Start Tx	FIFO EF	FIFO FF	N/U	VME _SRC	ATN Err	DUH WTM	MTU NS	BUSY	MDTB Err	Tx Mode	Tx Src

Figure 5-2: Control Register Bit Assignment

Reading of the data transfer mode bits (D0,D1,D7) provide information as to the direction of data transfers.

Error flags are provided to indicate improper transfer performance.

One bit is used to perform transfer initialization. This is the Start Transfer bit. When set, the interface specified by the mode control bits initiates transfer procedures.

The FIFO operations can be monitored via two status bits (Full and Empty).

When a power-up or software reset is given to the Tape Interface Board, the Control Register is placed in a known state: D0 to D5, D7, D10 to D12 are cleared. D9 goes to a logic high.

The following sections describe in details the operations of these controls.

5.1.2.1 Data Transfer Mode (D0, D1, D7)

The Data Transfer Mode bits provide control and monitoring of the data transfers accomplished by the Tape Interface Board. They are set via VMEbus and can be read at any time to inform on the direction of transfers. Table 5-3 shows the functions of these two bits.

Bit	Function
0	Transfer Source (TX_SRC): 0- Disk Unit 1- Tape Unit
1	Transfer Mode (TX_MODE): 0-Disk/Tape 1- VMEbus Monitoring
7	VMEbus to Tape Mode (VME_SRC): 0- No 1- Yes (disable other modes)

Table 5-3: Data Direction Control Bits

These three bits allow five modes of operation:

- Disk to tape transfers: TX_SRC = 0, TX_MODE = 0, VME_SRC = 0
- Tape to disk transfers: TX_SRC = 1, TX_MODE = 0, VME_SRC = 0
- VMEbus monitoring of disk data: TX_SRC = 0, TX_MODE = 1, VME_SRC = 0
- VMEbus monitoring of tape data: TX_SRC = 1, TX_MODE = 1, VME_SRC = 0
- VMEbus to Tape transfers: TX_SRC = "don't care", TX_MODE = "don't care", VME_SRC = 1

VMEbus monitoring is accomplished on 256 Kbyte blocks.

5.1.2.2 Error Detection (D2-D6)

Error detection is provided to indicate improper transfers taking place. These conditions are detected in various locations depending on their effect on the system.

Table 5-4 shows the assignment of these bits in the Control Register.

Bit	Function
2	Metrum Data Transfer Block Error: 0-No Error 1-Error
3	BUSY0-: 0- BUSY0- is not asserted; 1- BUSY0- is asserted.
4	Metrum Tape Unit Not Selected: 0- No Error 1- Metrum Unit Not Selected
5	Disk Unit Has Wrong Transfer Mode: 0- BLOCK Signal Not Asserted 1- BLOCK Signal Asserted
6	Metrum Tape Attention Error 0- ATTN-0 not asserted 1- ATTN-0 asserted

Table 5-4: Error Bit Assignment

Metrum Data Transfer Block Error (D2) - One source of error is the detection of semaphores from the Metrum data interface (Error-1, Error-2). The assertion of either signals indicates an error in the operation of the tape. On error detection, the bit (D2) is set to indicate an error. The bit is cleared by reading the Control Register.

BUSY0- (D3) - The BUSY0- monitor indicates the state of the BUSY0- signal on the tape unit command port. When read high, BUSY0- is asserted (BUSY0- =0). If read low, the BUSY0- signal is un-asserted (BUSY0- = 1). Whe asserted, this signal indicates that the tape unit is occupied at some task, sually command related. See the Metrum Tape Technical Manual [3] for more details.

Metrum Tape Unit Not Selected (D4) - Issuing a command to a tape unit (in this case the only one) which is not selected will cause the Metrum Tape Unit Not Selected (D4) bit to be set. The bit is cleared after reading the Control Register.

Disk Unit Has Wrong Transfer Mode (D5) - Another error can be detected on the Differential Fast Bus if the master (in this case the Tape Interface Board) detects an invalid state on the BLOCK signal. Transfers occurring while the BLOCK bit is set on the Differential Fast Bus, sets the Disk Unit Has Wrong Transfer Mode (D5) bit. The bit is cleared when reading the Control Register.

Metrum Tape Attention Flag (D6) - The Metrum tape unit signals problems by asserting a line on the tape command port. The line is called ATTN-0. A bit in the Control Register is used to monitor this line. If detected high ("1"), bit D6 of the register indicates the ATTN-0 line is asserted. The tape Attention Status register should be read. If the bit reads low, ATTN-0 is un-asserted and no action is required. Refer to the Metrum Tape Technical Manual [3] for more details.

5.1.2.3 FIFO Status (D9, D10)

The FIFO Status bits return information as to the state of data in the buffer (see Table 5-5). The FIFO Full flag indicates, when set, that no more data can be transferred into the buffers. When cleared, the bit indicates that more data can be input into the buffers.

The FIFO Empty Flag indicates when read that the data buffers are completely empty. If the buffers contain any data, the bit is cleared. This bit will be set following a FIFO reset.

Bit	Function
9	FIFO Full Flag: 0- FIFO Not Full, Able To Accept More Data. 1- FIFO Completely Full, No More Data Accepted.
10	FIFO Empty Flag: 0- FIFO Contains Data. 1- FIFO Completely Empty of Data.

Table 5-5: FIFO Status Bit Assignment

5.1.2.4 Start Transfer (D11)

The Start Transfer bit is used to initiate the transfer procedures. Table 5-6 defines the function of this bit in the Control Register. For disk to tape transfers, the disk interface indicates to the disk controls that the Tape Interface Board is ready to accept data from the disk array.

Bit	Function
11	Data Transfer Start: Reading: 0- Tape Interface Board Idle, No Transfers are Taking Place (Transfers Complete). 1- Tape Interface Board Transfers in Progress. Writing: 0- No Action. 1- Start Transfers, Wait for first data.

Table 5-6: Start Transfer Bit

For tape to disk transfers, the Differential Fast Bus interface indicates to the tape unit that transfers are ready to be accepted from the tape.

Transfers occur until no more data needs to be read. At the end of data transfers, the Start Transfer bit is reset, indicating, when read, that new transfers can be accomplished.

In VMEbus monitoring mode, this bit is used to initiate the transfer cycle. When a block of data has been captured, the bit is de-asserted.

5.1.2.5 Differential Fast Bus Master Enable (D12)

This bit enables the Tape Interface Board to assume control of the Differential Fast Bus. Before enabling this bit, the programmer must ensure that all other masters have been removed from the bus. After transfers are complete, the program must reset this bit to disable the Differential Fast Bus master on the Tape Interface Board (Table 5-7).

Bit	Function
12	Differential Fast Bus Master Enable: 0- Differential Fast Bus Master Disabled, All control and data lines in High Impedance. 1- Differential Fast Bus Master Enabled, All Data and Control Lines are Monitored or Driven.

Table 5-7: DFB Master Enable Bit

5.1.2.6 Unused Bits (D8, D13-D15)

These bit are reserved for possible enhancements.

5.1.3 Tape Command Port (x00002)

The Tape Command port issues commands to the tape unit. It operates in write mode (sending data to the tape unit) except when reading a Status register on the tape unit. The port issues 8 bits (LSB) of data to the tape interface. The 8 MSB's are ignored. Figure 5-3 shows the bit assignment for this register.

D15-D8	D7	D6	D5	D4	D3	D2	D1	D0
XXX	MSB	D6	D5	D4	D3	D2	D1	LSB

Figure 5-3: Tape Command Port

When using the Status Read command, the proper data sequence must be followed (see Metrum VLDS Manual for more details). Failure to do so will result in indeterminate operation of the tape drive. Following a Status Read command, reading the Tape Command Port will return the value issued by the tape unit (8 LSB's).

5.1.4 FIFO Reset Register (x00004)

The FIFO Reset register is used to reset the FIFO buffer under program control. Figure 5-4 shows the bit assignment for this register.

D15-D1	D0
XXX	FIFO Reset

Figure 5-4: FIFO Reset Register

To reset the FIFO, a "1" must be written to bit D0 of the register. Following reset, the FIFO buffers are cleared and the empty flag is set (section 3.5.2.3). In order to properly reset the buffers, write operation will be stopped. If a FIFO reset operation is performed during data transfers, the transfer is aborted before the reset is applied to the FIFO memories. To resume transfer operations, the whole process must be restarted from the beginning.

5.1.5 Tape Interface Board Reset Register (x00006)

This register allows the initializing of the complete card under program control. Figure 5-5 shows the bit assignment for this register.

D15-D5	D4	D3	D2	D1	D0
XXX	DFB	VLDS	TIB	0	0

Figure 5-5: TIB Reset Register

The register is designed to allow also for individual initialization of the disk or tape interface. Refer to Table 5-8 for register bit definitions. D1 and D0 are used to ensure correct register access. If D0 and D1 are not "0", the reset will not operate.

To reset the whole board, only bit D2 need be written high (logic "1"). D3 and D4 are "don't care". To reset individual interfaces, D2 must be written low (logic "0") and the individual bit set to a logic high. If D3 and/or D4 are written high (Logic "1"), only the interfaces are reset, FIFO buffers are left untouched. To reset FIFO buffers, see section 3.5.4. In all cases, D0 and D1 must be written low (logic "0"). Reading the register returns the state of the interfaces. A low level on any bit indicates inactivity on that particular module.

Bit	Function
D2	Tape Interface Board Reset: Writing: 0- No Action 1- Complete Tape Interface Board Reset.
D3	VLDS Interface Reset: Writing: 0- No Action 1- Places the Tape Interface in Idling, All Signals Unasserted.
D4	Differential Fast Bus Reset: Writing: 0- No Action 1- Places the DFB into High Impedance.

Table 5-8: TIB Reset Register Bit Definition

6. SOFTWARE DESCRIPTION

The integration of the Metrum tape unit into EARS is supported by means of functions incorporated into the System Controller multi-tasking program. Each function fulfills one specific operation. Most of these functions are under the control of the Data Acquisition window on the console. The others are initialization functions automatically called upon system start-up. All the functions have been added to *Acq_Data*, which is the task within the system controller program responsible for the data acquisition.

The functions added to *Acq_Data*, and the operations they perform, are:

- *ResetTibVlds():* Sends the TIB and the Metrum into reset
- *InitTib():* Initializes the TIB
- *InitVlds():* Initializes the Metrum tape unit
- *FormatVlds():* Formats a Metrum cassette
- *SkipVlds():* Skips filemarks on a Metrum cassette
- *RewindVlds():* Rewinds a Metrum cassette
- *EjectVlds():* Ejects a Metrum cassette
- *BkupC51ToVlds():* Transfers files from the C51 disk array to the Metrum tape unit
- *BkupVldsToC51():* Transfers files from the Metrum tape unit to the C51 disk array
- *BkupVldsToExa():* Transfers files from the Metrum tape unit to the Exabyte tape unit

Four utility functions are provided to ease the access to the Metrum command and status registers from the higher-level functions. They are:

- *VldsCassette():* Verifies if there is a cassette in the Metrum
- *VldsWritable():* Checks if the Metrum cassette is writable
- *SendVldsCmd():* Sends a command to the Metrum
- *ReadVldsStatus():* Reads the content of a Metrum status register

6.1 TIB AND METRUM RESET

When the System Controller program is invoked, it is safe to reset the TIB and the Metrum to make sure they run from a known state. This operation is performed by the *ResetTibVlds()* function. *ResetTibVlds()* is called upon starting the *Acq_Data* task, from the task's *main()* function.

ResetTibVlds() does a reset by setting both the TIB and the VLDS bits of the TIB Reset Register (described in section 5.1.5).

6.2 TIB INITIALIZATION

The TIB needs to be initialized before any access to the C51 or to the VLDS is initiated. This operation is performed by the *InitTib()* function.

Three functions use the TIB to access the C51 or the VLDS: *BkupC51ToVlds()*, *BkupVldsToC51()* and *BkupVldsToExa()*. In these functions, *InitTib()* is called before the data transfer is initiated. *InitTib()* is responsible for clearing the START_TX bit of the TIB Control Register (described in section 5.1.3).

6.3 METRUM INITIALIZATION

The Metrum is initialized upon starting the System Controller program, just after the TIB initialization. This ensures that the tape unit is in a well-defined state before using it. The operation is performed by *InitVlds()*, which is called from the *main()* function of the *Acq_Data* task.

The only operation performed by *InitVlds()* is to put the Metrum in the selected mode, by calling *SendVldsCmd()* with the SELECT_UNIT_0 parameter (hex 5A). This command puts the Metrum on-line, ready for operation. For a description of *SendVldsCmd()*, refer to section 6.13.

6.4 FORMAT A CASSETTE

The Format command allows to format a Metrum cassette. This command is performed when the EARS operator clicks the FORMAT button in the Data Acquisition window.

In the system controller program, the command is performed by *FormatVlds()*. This function is invoked by the *ProADMsg()* function, upon reception of the command from the console. To achieve its purpose, *FormatVlds()* performs the following operations:

1) **Verify the presence of a cassette.** This is done by calling *VldsCassette()*. If there is no cassette in the unit, a failure is signaled to the console and the function ends. *VldsCassette()* is described in section 6.11.
2) **Check if the cassette is writable.** This is done by calling *VldsWritable()*. If the cassette is write-protected, a failure is signaled to the console and the function ends. *VldsWritable()* is described in section 6.12.
3) **Tell the Metrum to start formatting the cassette**, by calling *SendVldsCmd()* with the FORMAT_WITH_ZERO_VOLUME_LABEL parameter (hex 50). This command tells the Metrum to format the cassette with the volume label "0". For a description of *SendVldsCmd()*, refer to section 6.13.
4) **Verify if the cassette has been successfully formatted.** The Metrum's Cassette Status register (TAPE_ALREADY_FORMATTED bit) is checked using *ReadVldsStatus()*, to verify if the cassette was already formatted. The result, success or failure, is forwarded to the console. For a description of *ReadVldsStatus()*, refer to section 6.14.

6.5 SKIP FILEMARKS ON A METRUM CASSETTE

The Skip Filemark command allows to skip the requested number of filemarks, in order to position the cassette at the beginning of a specific file. This utilization takes into account that each file is spaced by one filemark from its neighbors. The filemarks are skipped in either forward or reverse direction, depending if the number of filemarks requested is positive or negative.

The Skip Filemark command is performed when the EARS operator clicks the SKIP button in the Data Acquisition window. In the system controller program, the command is performed by *SkipVlds()*. This function is invoked by the *ProADMsg()* function, upon reception of the command from the console. To achieve its purpose, *SkipVlds()* performs the following operations:

1) **Verify the presence of a cassette.** This is done by calling *VldsCassette()*. If there is no cassette in the unit, a failure is signaled to the console and the function ends. *VldsCassette()* is described in section 6.11.
2) **Tell the Metrum to skip the requested number of filemarks.** On the Metrum, two commands are available: SEARCH_FILEMARK_FORWARD (hex 4B) and SEARCH_FILEMARK_REVERSE (hex 4C). Both of them can only skip one filemark at a time. Therefore they are repeatedly sent for each filemark to skip, using *SendVldsCmd()*. For a description of *SendVldsCmd()*, refer to section 6.13.
3) **Verify if each requested filemark has been found.** This covers the detection of beginning-of-tape, end-of-tape or end-of-data . The Metrum's System Status I register (SEARCH_ERROR bit) is checked for that purpose, using *ReadVldsStatus()*. The result, success or failure, is forwarded to the console. For a description of *ReadVldsStatus()*, refer to section 6.14.

6.6 REWIND A METRUM CASSETTE

The Rewind command allows to fully rewind a cassette at the maximum speed. This command is performed when the EARS operator clicks the REWIND button in the Data Acquisition window.

In the system controller program, the command is performed by *RewindVlds()*. This function is invoked by the *ProADMsg()* function, upon reception of the command from the console. To achieve its purpose, *RewindVlds()* performs the following operations:

1) **Verify the presence of a cassette.** This is done by calling *VldsCassette()*. If there is no cassette in the unit, a failure is signaled to the console and the function ends. *VldsCassette()* is described in section 6.11.
2) **Tell the Metrum to rewind the cassette,** by calling *SendVldsCmd()* with the FAST_REVERSE parameter (hex 44). For a description of *SendVldsCmd()*, refer to section 6.13.

6.7 EJECT A METRUM CASSETTE

The Eject command allows to eject a cassette from the tape unit. This command is performed when the EARS operator clicks the EJECT button in the Data Acquisition window.

In the system controller program, the command is performed by EjectVlds(). This function is invoked by the ProADMsg() function, upon reception of the command from the console. To achieve its purpose, EjectVlds() performs the following operations:

1) **Verify the presence of a cassette.** This is done by calling *VldsCassette()*. If there is no cassette in the unit, a failure is signaled to the console and the function ends. *VldsCassette()* is described in section 6.11.
2) **Tell the Metrum to go in the Subload mode.** The Metrum can eject a cassette only when in the subload mode. The mode is sent by calling *SendVldsCmd()* with the SUBLOAD parameter (hex 42).
3) **Tell the Metrum to eject the cassette,** by calling *SendVldsCmd()* with the EJECT parameter (hex 41). For a description of *SendVldsCmd()*, refer to section 6.13.

6.8 TRANSFER FROM DISK ARRAY TO METRUM

This command allows to back-up one file from the C51 disk array to the Metrum tape unit. It is invoked from the Data Acquisition window, when the EARS operator clicks the BACKUP button with the C51 and the Metrum selected as source and target devices, respectively. The back-up is performed for all the high-lighted, selected files. If more than one file is selected, the command is invoked sequentially for each of them.

In the system controller program, the command is performed by *BkupC51ToVlds()*. The file index is passed as a parameter to the function. To achieve its purpose, *BkupC51ToVlds()* performs the operations descibed below. The flow of operations is given in Figure 6-1.

PREPARE ALL THE DEVICES:

1) **Verify the presence of a cassette.** This is done by calling *VldsCassette()*. If there is no cassette in the unit, a failure is signaled to the console and the function ends. *VldsCassette()* is described in section 6.11.
2) **Check if the cassette is writable.** This is done by calling *VldsWritable()*. If the cassette is write-protected, a failure is signaled to the console and the function ends. *VldsWritable()* is described in section 6.12.
3) **Initialize the TIB.** This is done by calling *InitTib()*, which is described in section 6.2.
4) **Set the TIB for C51-to-Metrum data transfers,** by initializing the TRANSFER_SOURCE and TRANSFER_MODE bits of the TIB Control Register. Refer to section 5.1.3 for a description about those bits.

5) **Tell the C51 to operate in the continuous mode, and the VMEHD to release the DFB.** This operation tells the C51 to use the continuous mode to transfer its data on the DFB, in order to be compatible with the TIB. It also orders the VMEHD controller board to disable its DFB port, in order for an external device (the TIB) to communicate with the C51. All these operations are done by calling *C51Config()*.

6) **Set the DFB master mode on the TIB.** The TIB must be the master during the data transfer from the C51. This is done by initializing the TIB Control register's DFB_MASTER_ENABLE bit. Refer to section 5.1.3 for a description about that bit.

7) **Position the C51 at the file beginning,** to ensures that the disk is ready to transfer the requested file. The operation is done by calling *C51Seek()*.

8) **Tell the C51 to output the file.** This operation tells the C51 the amount of data to read from its disks (i.e. the file size). The operation is done by calling *C51Read()*.

TRANSFER THE FILE TO THE C51:

9) **Tell the Metrum to record data.** This operation initiates the data transfer from the C51 to the Metrum, until the whole file is transferred. The operation is done by using *SendVldsCmd()* to send the RECORD command (hex 46) to the Metrum.

10) **Tell the TIB to start the transfer,** by initializing the START_TX bit of the TIB Control Register. Refer to section 5.1.3 for a description about that bit.

11) **Wait until the whole file is written to tape,** by monitoring the START_TX bit of the TIB Control Register.

AT THE END OF THE FILE TRANSFER:

12) **Remove the DFB master mode on the TIB,** by initializing the DFB_MASTER_ENABLE bit of the TIB Control Register. This ensure that other devices will be able to use the DFB.

13) **Inform the console** that the file back-up is done.

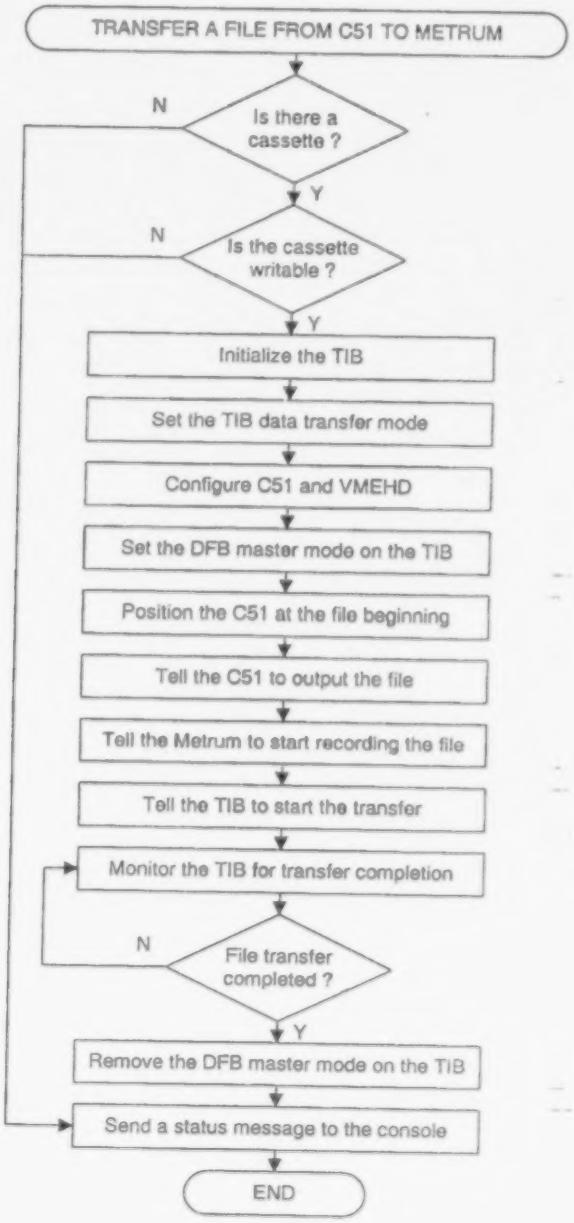

TRANSFER A FILE FROM C51 TO METRUM
Is there a cassette ?
N
Y
Is the cassette writable ?
N
Y
Initialize the TIB
Set the TIB data transfer mode
Configure C51 and VMEHD
Set the DFB master mode on the TIB
Position the C51 at the file beginning
Tell the C51 to output the file
Tell the Metrum to start recording the file
Tell the TIB to start the transfer
Monitor the TIB for transfer completion
File transfer completed ?
N
Y
Remove the DFB master mode on the TIB
Send a status message to the console
END

Figure 6-1: Disk Array to Metrum Transfer Algorithm

6.9 TRANSFER FROM METRUM TO DISK ARRAY

This command allows to back-up all the files of a Metrum cassette to the C51 disk array. It is invoked from the Data Acquisition window, when the EARS operator clicks the BACKUP button with the Metrum and the C51 selected as source and target devices, respectively. The back-up is performed one file at a time from the tape's current position. The file is written on the disk, at the end of the last recorded file.

In the system controller program, the command is performed by *BkupVldsToC51()*. The function requires no parameters. To achieve its purpose, *BkupVldsToC51()* performs the operations descibed below. The flow of operations is given in Figure 6-2.

PREPARE ALL THE DEVICES:

1) **Verify the presence of a cassette.** This is done by calling *VldsCassette()*. If there is no cassette in the unit, a failure is signaled to the console and the function ends. *VldsCassette()* is described in section 6.11.

2) **Initialize the TIB.** This is done by calling *InitTib()*, which is described in section 6.2.

3) **Set the TIB for Metrum-to-C51 data transfers,** by initializing the TRANSFER_SOURCE and TRANSFER_MODE bits of the TIB Control Register. Refer to section 5.1.3 for a description about those bits.

4) **Tell the C51 to operate in the continuous mode, and the VMEHD to release the DFB.** This operation tells the C51 to use the continuous mode to transfer its data on the DFB, in order to be compatible with the TIB. It also orders the VMEHD controller board to disable its DFB port, in order for an external device (the TIB) to communicate with the C51. All these operations are done by calling *C51Config()*.

5) **Set the DFB master mode on the TIB.** The TIB must be the master during the data transfer from the C51. This is done by initializing the TIB Control register's DFB_MASTER_ENABLE bit. Refer to section 5.1.3 for a description about that bit.

6) **Position the C51 at the end of the last recorded file.** This operation is done by calling *C51Seek()*.

TRANSFER EACH FILE ON THE CASSETTE TO THE C51:

7) **Tell the Metrum to reproduce one file.** This is done by using *SendVldsCmd()* to send the REPRODUCE_N_FILEMARKS command (hex 6C) to the Metrum, with the number of filemarks "N" equal to 1.

8) **Verify if there is another file to transfer,** by using *ReadVldsStatus()* to check the CONTROL_TRACK_FAULT bit of the Metrum's Cassette Status register. This bit indicates the end-of-data and end-of-tape events. If one of these conditions is detected, it means that there is no more files left to transfer. When the event occurs, it is signalled to the console and the function ends.

9) **Tell the C51 to input the file.** This operation tells the C51 the amount of data to write on its disks. The operation is done by calling *C51Write()*. Since the size of the files on the Metrum is unknown, the file size sent to the C51 is the total remaining disk space.

10) **Tell the TIB to start the transfer,** by initializing the START_TX bit of the TIB Control Register. Refer to ??? for a description about that bit.

11) **Wait until the whole file is written to disk,** by monitoring the START_TX bit of the TIB Control Register.

12) **Tell the C51 to stop writing data, and inform the console that a file has been backed-up to the C51.** This is done by calling *C51Abort()*. At this point, the function has completed the transfer of one file, and it is ready to transfer another one.

AT THE END OF THE FILE TRANSFER:

13) **Remove the DFB master mode on the TIB,** by initializing the DFB MASTER_ENABLE bit of the TIB Control Register. This ensure that other devices will be able to use the DFB.

14) **Inform the console** that the back-up is done.

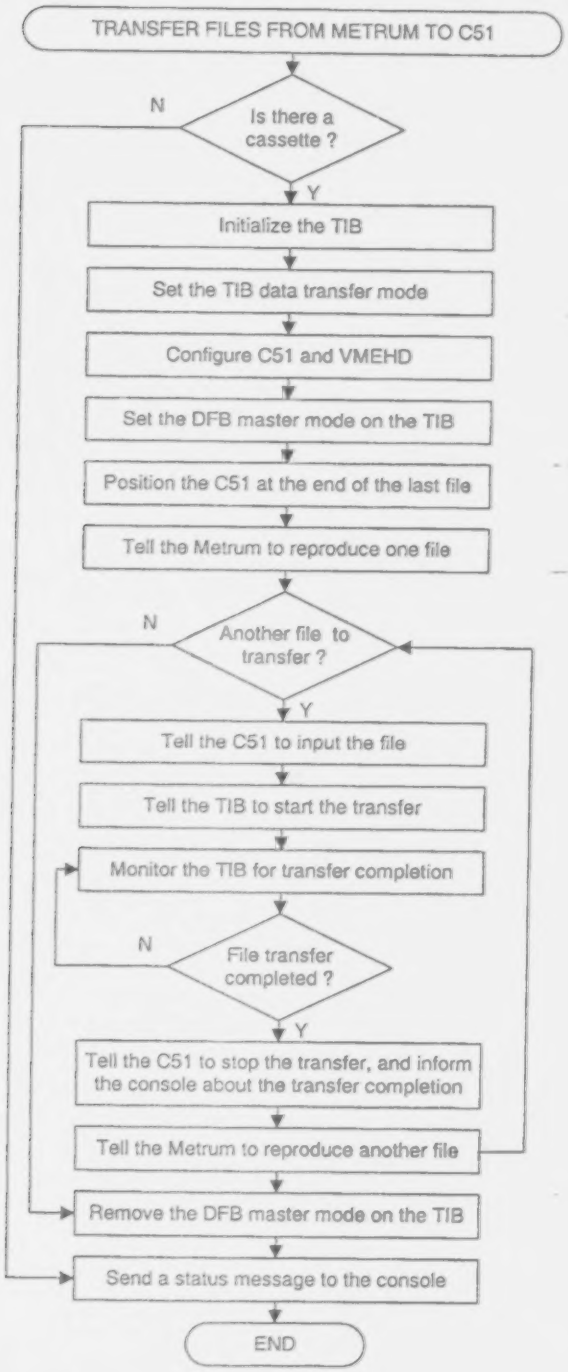

TRANSFER FILES FROM METRUM TO C51
Is there a cassette ?
N
Y
Initialize the TIB
Set the TIB data transfer mode
Configure C51 and VMEHD
Set the DFB master mode on the TIB
Position the C51 at the end of the last file
Tell the Metrum to reproduce one file
Another file to transfer ?
N
Y
Tell the C51 to input the file
Tell the TIB to start the transfer
Monitor the TIB for transfer completion
File transfer completed ?
N
Y
Tell the C51 to stop the transfer, and inform the console about the transfer completion
Tell the Metrum to reproduce another file
Remove the DFB master mode on the TIB
Send a status message to the console
END

Figure 6-2: Metrum to Disk Array Transfer Algorithm

6.10 TRANSFER FROM METRUM TO EXABYTE

This command allows to back-up all the files of a Metrum cassette to an Exabyte cassette. It is invoked from the Data Acquisition window, when the EARS operator clicks the BACKUP button with the Metrum and the Exabyte selected as source and target devices, respectively. The transfer is accomplished from the current position of both tapes, one file at a time. Each file is transferred by blocks from the Metrum to the CPU board through VME, and then from the CPU board to the Exabyte through SCSI. The block size used to transfer files is given by the size of the TIB buffers.

In the system controller program, the command is performed by BkupVldsToExa(). The function requires no parameter. To achieve its purpose, BkupVldsToExa() performs the operations described below. The flow of operations is given in Figure 6-3.

PREPARE ALL THE DEVICES:

1) **Verify the presence of a cassette.** This is done by calling *VldsCassette()*. If there is no cassette in the unit, a failure is signaled to the console and the function ends. *VldsCassette()* is described in section 6.11.

2) **Open a path to the Exabyte.** This is done by calling the *open()* routine of the standard C library. If there is a failure opening the path (no cassette or write-protected cassette), it is signaled to the console and the function ends.

3) **Initialize the TIB.** This is done by calling *InitTib()*, which is described in section 6.2.

4) **Set the TIB for Metrum-to-VME data transfers,** by initializing the TRANSFER_SOURCE and TRANSFER_MODE bits of the TIB Control Register. Refer to section 5.1.3 for a description about those bits.

TRANSFER EACH FILE FROM THE METRUM CASSETTE TO THE EXABYTE CASSETTE:

5) **Tell the Metrum to reproduce one file.** This is done by using *SendVldsCmd()* to send the REPRODUCE_N_FILEMARKS command (hex 6C) to the Metrum, with the number of filemarks "N" equal to 1.

6) **Verify if there is another file to transfer,** by using *ReadVldsStatus()* to check the CONTROL_TRACK_FAULT bit of the Metrum's Cassette Status register. This bit indicates the end-of-data and end-of-tape events. If one of these conditions is detected, it means that there is no more files left to transfer. When the event occurs, it is signalled to the console and the function ends.

TRANSFER EACH 128K BLOCKS OF THE FILE:

7) **Tell the TIB to read a file block from the Metrum,** by initializing the START_TX bit of the TIB Control Register. Refer to section 5.1.3 for a description about this bits.

8) **Wait until the block is ready,** by monitoring the START_TX bit of the TIB Control Register.

9) **Transfer the block from the TIB to the Exabyte**, using the *write()* routine of the standard C library.

10) **Verify if the whole file has been read from the Metrum**, by using *ReadVldsStatus()* to check the BUSY bit of the Metrum's System Status I register. If the file has been totally read, the Exabyte is told to write a filemark, and the function tries to transfer another file; otherwise, the function tries to transfer another block of the file.

AT THE END OF THE FILE TRANSFER:

11) **Inform the console** that the back-up is done.

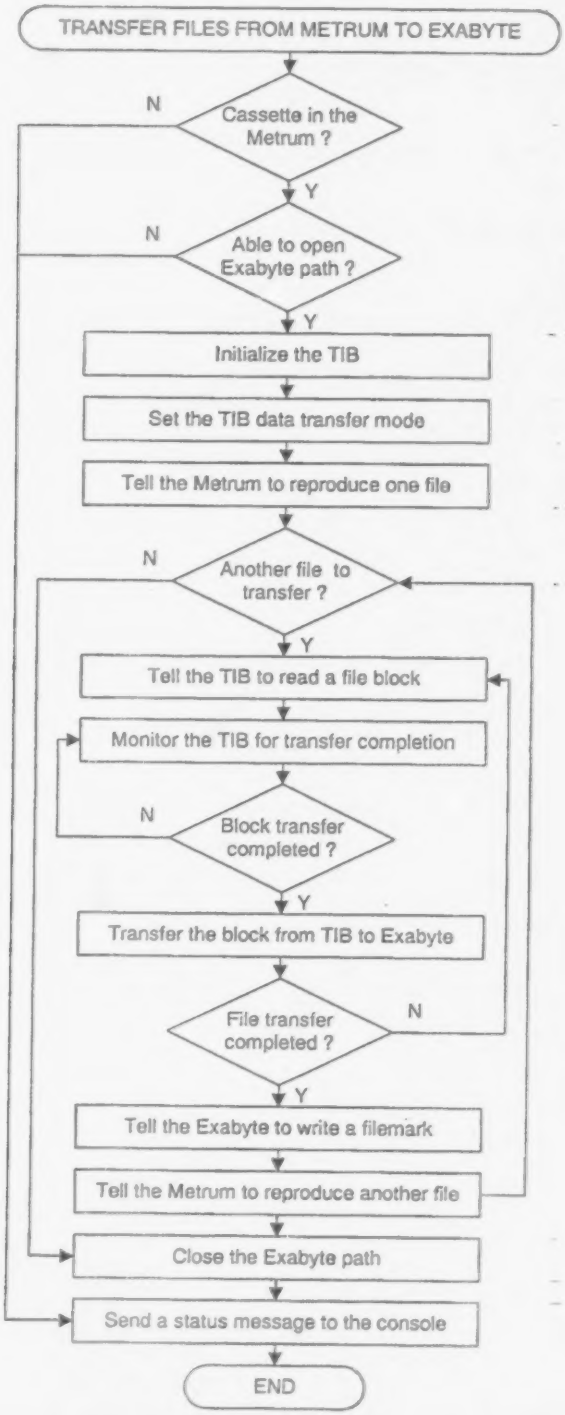


Figure 6-3: Metrum to Exabyte Transfer Algorithm

6.11 VERIFY THE PRESENCE OF A CASSETTE

VldsCassette() is a utility function allowing higher-level function to easily verify if there is a cassette in the Metrum. The function requires no parameters.

VldsCassette() finds out if there is a cassette in the unit, by calling *ReadVldsStatus()* to read the Metrum's Special Status register, specifically the ELEVATOR_UP and ELEVATOR_DOWN bits. If there is a cassette, TRUE is returned to the calling function; otherwise, FALSE is returned. For a description of *ReadVldsStatus()*, refer to section 6.14.

6.12 CHECK IF THE CASSETTE IS WRITABLE

VldsWritable() is a utility function allowing higher-level function to easily verify if the Metrum cassette is write-protected or not. The function requires no parameters.

VldsWritable() finds out if the cassette is writable, by calling *ReadVldsStatus()* to read the Metrum's Cassette Status register, specifically the WRITE_PROTECT bit. If cassette is not write-protected, TRUE is returned to the calling function; otherwise, FALSE is returned. For a description of *ReadVldsStatus()*, refer to section 6.14.

6.13 SEND COMMANDS TO THE METRUM

SendVldsCmd() is a utility function allowing higher-level function to easily send commands to the Metrum. The function receives the command to send through a pointer to an array of bytes. The first byte of the array is the command opcode. The second byte is the number of parameters to follow. The other bytes are the parameter accompanying the opcode.

SendVldsCmd() operates in two steps. First the function writes the received command and parameters to the TIB's Tape Command Port, to the attention of the Metrum tape unit.

Then the function waits for the Metrum to complete the command execution by monitoring the Metrum's BUSY bit. The BUSY bit is accessed by reading the Metrum's *System Status I* register, using *ReadVldsStatus()*. The waiting loop ensures that the current command will be completed before inquiring about its success or sending another command. To ensure that no CPU time is wasted in the loop, the *sleep()* routine of the standard C library is periodically called. Once the BUSY bit has been de-asserted, *SendVldsCmd()* ends.

For a description of *ReadVldsStatus()*, refer to section 6.14.

6.14 READ STATUS FROM THE METRUM

ReadVldsStatus() is a utility function allowing higher-level function to easily read any of the Metrum status registers. The function receives the address of the register to read as a one-byte parameter.

ReadVldsStatus() operates in two steps. First the function writes the STATUS_READ command (hex 51) and the requested register's address to the TIB's Tape Command port, to the attention of the Metrum tape unit. The register address is written in the form of two ASCII decimal bytes.

Then the function reads the TIB's Tape Command port, to obtain the one-byte content of the register. This ends the function.

7. REFERENCES

[1] *Development of an Enhanced Signal Processing Data Acquisition and Display System*, MultiDev JDL Communications Inc., June 1995.
[2] *Differential Fast Bus Interface Manual*, revision 2.0, Storage Concepts, May 1989.
[3] *Metrum Technical Manual*, Metrum, February 1994.

8. CONNECTOR PIN-OUT

8.1 TAPE DATA PORT

2x40 Pin Ribbon (3M #3432-L002)

Channel 1 Data	
1- Spare	2- GND
3- Record Ready	4- GND
5- Studio Sync	6- GND
7- GND	8- GND
9- D7	10- GND
11- D6	12- GND
13- D5	14- GND
15- Read Data Strobe	16- GND
17- Clock In	18- GND
19- D4	20- GND
21- D3	22- GND
23- D2	24- GND
25- Write Data Strobe	26- GND
27- Error 1	28- GND
29- Clock Out	30- GND
31- D1	32- GND
33- D0	34- GND
35- Spare	36- GND
37- Spare	

Channel 2 Data	
1- Direct Channel In	2- Direct Channel In RTN
3- Direct Channel Out	4- Direct Channel Out RTN
5- Spare	6- Spare
7- Spare	8- Spare
9- D15	10- GND
11- D14	12- GND
13- D13	14- GND
15- E-E Strobe	16- GND
17- Spare	18- Spare
19- D12	20- GND
21- D11	22- GND
23- D10	24- GND
25- Spare	26- Spare
27- Error 2	28- GND
29- Spare	30- Spare
31- D9	32- GND
33- D8	34- GND
35- Spare	36- Spare
37- Spare	

8.2 TAPE CONTROL PORT

64 Pin Ribbon (3M #3338-000)

IN

P2-1a	CD0	P2-1c	SPARE
P2-2a	CD1	P2-2c	SPARE
P2-3a	CD2	P2-3c	SPARE
P2-4a	CD3	P2-4c	GND
P2-5a	CD4	P2-5c	GND
P2-6a	CD5	P2-6c	GND
P2-7a	CD6	P2-7c	GND
P2-8a	CD7	P2-8c	GND
P2-9a	STB-0	P2-9c	GND
P2-10a	IBF-0	P2-10c	SELECT-0
P2-11a	OBF-0	P2-11c	ATTN-0
P2-12a	ACK-0	P2-12c	RESET-0
P2-13a	BUSY-0	P2-13c	RESET-0

OUT

P2-14a	CD0	P2-14c	SPARE
P2-15a	CD1	P2-15c	SPARE
P2-16a	CD2	P2-16c	SPARE
P2-17a	CD3	P2-17c	GND
P2-18a	CD4	P2-18c	GND
P2-19a	CD5	P2-19c	GND
P2-20a	CD6	P2-20c	GND
P2-21a	CD7	P2-21c	GND
P2-22a	STB-0	P2-22c	GND
P2-23a	IBF-0	P2-23c	SELECT-0
P2-24a	OBF-0	P2-24c	ATTN-0
P2-25a	ACK-0	P2-25c	RESET-0
P2-26a	BUSY-0	P2-26c	RESET-0

8.3 DIFFERENTIAL FAST BUS

1-	GND	2-	N. C.
3-	RSTB+	4-	RSTB-
5-	IORDY+	6-	IORDY-
7-	WSTB+	8-	WSTB-
9-	P16_LO+	10-	P16_LO-
11-	DB00+	12-	DB00-
13-	DB01+	14-	DB01-
15-	DB02+	16-	DB02-
17-	DB03+	18-	DB03-
19-	DB04+	20-	DB04-
21-	DB05+	22-	DB05-
23-	DB06+	24-	DB06-
25-	DB07+	26-	DB07-
27-	DB08+	28-	DB08-
29-	DB09+	30-	DB09-
31-	DB10+	32-	DB10-
33-	DB11+	34-	DB11-
35-	DB12+	36-	DB12-
37-	DB13+	38-	DB13-
39-	DB14+	40-	DB14-
41-	DB15+	42-	DB15-
43-	MRDY+	44-	MRDY-
45-	P16_HI+	46-	P16_HI-
47-	BLOCK+	48-	BLOCK-
49-	N. C.	50-	GND

8.4 VMEbus Connectors

(For P2 Rows A and C, see Tape Control Port Connector Pin-Out)

UNCLASSIFIED

SECURITY CLASSIFICATION OF FORM
(highest classification of Title, Abstract, Keywords)

DOCUMENT CONTROL DATA

(Security classification of title, body of abstract and indexing annotation must be entered when the overall document is classified)

1. ORIGINATOR (the name and address of the organization preparing the document. Organizations for whom the document was prepared, e.g. Establishment sponsoring a contractor's report, or tasking agency, are entered in section 8.)

MultiDev JDL Communications Inc.
1040 St.Joseph Blvd., suite 8
Hull, Quebec J8Z 1X9

2. SECURITY CLASSIFICATION (overall security classification of the document, including special warning terms if applicable)

UNCLASSIFIED

3. TITLE (the complete document title as indicated on the title page. Its classification should be indicated by the appropriate abbreviation (S, C or U) in parentheses after the title.)

Tape Interface Board - Detailed Design Document (U)

4. AUTHORS (Last name, first name, middle initial)

Blanchet, Pierre
Da Sylva, Pierre

5. DATE OF PUBLICATION (month and year of publication of document)

January 1996

6a. NO. OF PAGES (total containing information. Include Annexes, Appendices, etc.)

46 pages

6b. NO. OF REFS (total cited in document)

3

7. DESCRIPTIVE NOTES (the category of the document, e.g. technical report, technical note or memorandum. If appropriate, enter the type of report, e.g. interim, progress, summary, annual or final. Give the inclusive dates when a specific reporting period is covered.)

Technical Report, Design Document

8. SPONSORING ACTIVITY (the name of the department project office or laboratory sponsoring the research and development. Include the address.)

Defence Research Establishment Ottawa, Radar Division
3701 Carling Ave., Ottawa, Ontario

9a. PROJECT OR GRANT NO. (if appropriate, the applicable research and development project or grant number under which the document was written. Please specify whether project or grant.)

9b. CONTRACT NO. (if appropriate, the applicable number under which the document was written.)

W7714-4-9858

10a. ORIGINATOR'S DOCUMENT NUMBER (the official document number by which the document is identified by the originating activity. This number must be unique to this document.)

MD 002-04

10b. OTHER DOCUMENT NOS. (Any other numbers which may be assigned this document either by the originator or by the sponsor.)

11. DOCUMENT AVAILABILITY (any limitations on further dissemination of the document, other than those imposed by security classification.)

- (✓) Unlimited distribution
- ( ) Distribution limited to defence departments and defence contractors: further distribution only as approved
- ( ) Distribution limited to defence departments and Canadian defence contractors: further distribution only as approved
- ( ) Distribution limited to government departments and agencies: further distribution only as approved
- ( ) Distribution limited to defence departments: further distribution only as approved
- ( ) Other (please specify):

12. DOCUMENT ANNOUNCEMENT (any limitation to the bibliographic announcement of this document. This will normally correspond to the Document Availability (11). However, where further distribution (beyond the audience specified in 11) is possible, a wider announcement audience may be selected.)

Unlimited distribution